AD-A090 775 NIELSEN ENGINEERING AND RESEARCH INC MOUNTAIN VIEW CA F/G 1/1
AERUELASTIC OPTIMIZATION WITH MULTIPLE CONSTRAINTS.(U)
SEP 60 S C MCINTOSH F49620-78-C-0105
UNCLASSIFIED NEAR-TR-228 AFOSR-TR-80-1070 NL

1 OF 1
AD
A090775

END
DATE
FILMED
11-80
DTIC

12
sc

Approved for public release;
distribution unlimited.

NIELSEN ENGINEERING
AND RESEARCH, INC.

OFFICES: 510 CLYDE AVENUE / MOUNTAIN VIEW, CALIFORNIA 94043 / TELEPHONE (415) 968-9457

AEROELASTIC OPTIMIZATION WITH MULTIPLE CONSTRAINTS

by

S. C. McIntosh, Jr.

NEAR TR 228

September 1980

Prepared under Contract No. F49620-78-C-0105

for

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
Washington, D.C.

by

NIELSEN ENGINEERING & RESEARCH, INC.
510 Clyde Avenue, Mountain View, CA 94043
Telephone (415) 968-9457

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 18 AFOSR-TR-80-1070	2. GOVT ACCESSION NO. AD-A090775	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AEROELASTIC OPTIMIZATION WITH MULTIPLE CONSTRAINTS		5. TYPE OF REPORT & PERIOD COVERED Final Scientific Report 7/15/78 to 11/14/79
	14	6. PERFORMING ORG. REPORT NUMBER NEAR-TR-228
7. AUTHOR(s) 10 S. C. McIntosh, Jr.	15	8. CONTRACT OR GRANT NUMBER(s) F49620-78-C-0105
9. PERFORMING ORGANIZATION NAME AND ADDRESS Nielsen Engineering & Research 510 Clyde Avenue Mountain View, CA 20332	16	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2307/B1 61102F 17 B1
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research Building 410, Bolling AFB Washington, D.C. 20332	N/A 11	12. REPORT DATE Sep ~~tember 1980~~ 80
	12	13. NUMBER OF PAGES 54 58
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 9 Final scientific rept. 15 Jul 78-14 Nov 79		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Optimization; structural design; aeroelasticity

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Two optimality-criterion algorithms for treating problems with multiple behavioral constraints are described. The first employs slack variables to recast inequality constraints as equality constraints, so that the optimization procedure can be based on an earlier one developed for multiple equality constraints. The second is based on earlier work that uses Gauss-Seidel iteration to identify the active constraints by identifying the set of positive Lagrange multipliers. Both methods are applied to some simple test

384783 JOB

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

cases with two design variables, and then the slack-variable algorithm is applied to optimize the weight of a biconvex sandwich wing under various combinations of flutter and frequency constraint The behavior of this algorithm is discussed and recommendations are given for future work.

UNCLASSIFIED

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF FIGURES (Concluded)

AEROELASTIC OPTIMIZATION WITH MULTIPLE CONSTRAINTS

1. INTRODUCTION

This report summarizes the work performed under the sponsorship of the Air Force Office of Scientific Research, with Contract No. F49620-78-C-0105. The overall goal of this research has been to develop improved methods of designing least-weight aerospace structures that satisfy both strength and stiffness requirements.

During the first year, some efficiency comparisons were undertaken with a number of different optimization algorithms. All of these algorithms were applied to problems involving a single behavioral constraint. This work is summarized in reference 1. From this work, two alyorithms were chosen for work with multiple constraints. The first algorithm is an extension of work performed by Segenreich and co-authors (refs. 2-4). The idea here is to employ slack variables in order to transform the usual inequality behavioral constraints into equality constraints. In this manner, an optimization algorithm based on multiple equality constraints can be applied. Such an approach has been applied to problems with static constraints in reference 4.

The second algorithm was developed by Rizzi (ref. 5). This algorithm employs a Gauss-Seidel iterative procedure to aid in the elimination of inactive constraints. It has been applied by Rizzi in reference 5 to problems involving both strength and stiffness constraints, and it was felt desirable to use it for comparison with the first algorithm on larger problems. Since both of the algorithms discussed above employ a similar resizing formula, the comparison proposed is primarily a test of two different procedures for handling multiple constraints.

The sections that follow will describe in detail the two algorithms as coded here, the numerical applications carried out, and some recommendations for future work.

2. STATEMENT OF PROBLEM AND THEORETICAL BACKGROUND

Consider the problem of minimizing the total mass M_T of a structure. Let the design variables be t_i, $i = 1,2\ldots,M$, which may either be member-sizing variables in a finite-element model of the structure or concentrated-mass sizing variables. With the usual types of finite elements, the mass is a linear function of the design variables, and the total mass can be written as

$$M_T = M_o + \sum_{i=1}^{M} a_i t_i \tag{1}$$

where M_o is the mass not available for optimization, such as fuel mass. There are behavioral constraints on the structure, representing strength or stiffness requirements such as maximum allowable stresses, minimum frequencies of free vibration, maximum displacement, or minimum flutter speeds. These are characterized as inequality constraints

$$c_j(t_i) \leq 0, \quad j = 1,2,\ldots\ldots,N \tag{2}$$

Finally, there can be side constraints that define the maximum and minimum values of the design variables:

$$(t_i)_{min} \leq t_i \leq (t_i)_{max}, \quad i = 1,2,\ldots,M \tag{3}$$

Necessary conditions for a local minimum are given by the Kuhn-Tucker conditions. These can be written in the form

$$\bar{v}_i = a_i + \sum_{j=1}^{N} \lambda_j \frac{\partial c_j}{\partial t_i} = 0, \ (t_i)_{min} \leq t_i \leq (t_i)_{max} \tag{4}$$

$$\bar{v}_i \geq 0 \ , \quad t_i = (t_i)_{min} \tag{5}$$

$$\bar{v}_i \leq 0 \ , \quad t_i = (t_i)_{max} \tag{6}$$

with the multipliers λ_j given by

$$\lambda_j = 0 \quad \text{if } c_j < 0 \qquad (7)$$

$$\lambda_j \geq 0 \quad \text{if } c_j = 0 \qquad (8)$$

There are essentially two parts to solving an optimization problem as posed above. The first involves evaluating the constraints and computing derivatives for some design; the second involves a redesign algorithm to compute the next design change in an iterative process that leads to a local optimum. It has been customary in the past to distinguish between direct search methods (or mathematical-programming methods) and optimality-criterion methods for redesign. However, recent work (refs. 6 and 7) has shown that these differences in many cases are more philosophical than real. For present purposes, it is sufficient to note that the principal problem for large numbers of constraints, such as stress constraints on individual members, is to identify in an efficient manner that subset of the constraints that is active - i.e., that satisfies equation (8) - at optimum. Subsequent sections will describe in detail two strategies that have been proposed to deal with this problem.

3. AN OPTIMALITY-CRITERION METHOD EMPLOYING SLACK VARIABLES

3.1 General Description

The method as described here is an adaptation of that described in reference 4. The principal idea behind this method is to use slack variables to convert inequality constraints to equality constraints. The problem of identifying active and inactive constraints is avoided, and the redesign algorithm is somewhat simplified, since it need be set up only for equality constraints.

The inequality constraints given by equation (2) are written in the equivalent form

$$c_j'(t_i) = c_j(t_i) + z_i^2 = 0 \qquad (9)$$

These constraints are appended to the merit function (the total mass) in the form

$$M' = M_T + \sum_{j=1}^{N} \lambda_j c_j'$$

$$= M_o + \sum_{i=1}^{M} a_i t_i + \sum_{j=1}^{N} \lambda_j (c_j + z_j^2) \qquad (10)$$

Equations (4)-(6) remain unchanged, and equations (7) and (8) are replaced by

$$\lambda_j z_j = 0 \quad , \quad j = 1,2,\ldots,N \qquad (11)$$

$$c_j + z_j^2 = 0 \quad , \quad j = 1,2,\ldots,N \qquad (12)$$

Equations (11) and (12) can be interpreted to mean that when a constraint c_j is active at the optimum, then the corresponding slack variable z_j is zero, and λ_j can have an arbitrary value. Or, if the constraint is inactive, then z_j is not zero and λ_j must be zero to satisfy equation (11). Equations (4)-(6), (11), and (12) constitute $2N+M$ equations for $2N+M$ unknowns - the M design variables t_i, the N multipliers λ_j and the N slack variables z_j.

The redesign algorithms for the design variables and the slack variables are (ref. 4)

$$t_i^{\nu+1} = \begin{cases} C_i^\nu t_i^\nu \quad , & (t_i)_{min} \le C_i^\nu t_i^\nu \le (t_i)_{max} \\ (t_i)_{min}, & C_i^\nu t_i^\nu \le (t_i)_{min} \\ (t_i)_{max}, & C_i^\nu t_i^\nu \ge (t_i)_{max} \end{cases} \qquad (13)$$

$$z_j^{\nu+1} = D_j^\nu z_j^\nu \quad , \qquad (14)$$

where ν is the iteration number and

$$C_i^\nu = \alpha^\nu + (\alpha^\nu-1)\,\frac{1}{a_i}\sum_{j=1}^{N}\lambda_j\left(\frac{\partial c_j}{\partial t_i}\right)^\nu \qquad (15)$$

$$D_i^\nu = 1 + (\alpha^\nu-1)K\lambda_j^\nu \qquad (16)$$

The redesign factor C_i^ν is the same as that proposed by Kiusalaas (ref. 8). From the optimality criterion, equation (4), it can be seen that C_i^ν approaches unity, independently of the value of the relaxation parameter α^ν, as the optimum design is approached. From equation (11), it can be seen that the factor D_i^ν approaches unity in similar fashion as the design converges. The constant K is a free parameter that can be selected by the designer; in effect, it permits controlling the step size on the slack variables differently from the step size on the design variables.

To find the Lagrange multipliers λ_j, the requirement that $c_j' = 0$ for all j is imposed. This is equivalent to the procedure employed in an optimization problem with multiple equality constraints, except that there are some additional terms involving the slack variables. Also, for the sake of completeness, terms will be included to account for design variables that are reaching their upper or lower bounds. As a design variable reaches one of these bounds, it is left at the appropriate value and relegated to the passive set. After a converged solution has been obtained, tests based on equations (5) and (6) are applied to see if any passive variable is to be reintroduced to the active set.

Now let A_v^ν denote the active set of design variables, and assume that at the νth iteration equations (13) have been applied, and those design variables that are going into the passive set (<u>not</u> those relegated to the passive set from a previous iteration) have been identified. Let P_ℓ^ν and P_u^ν denote such design variables for the lower (or minimum-value) and upper (or maximum-value) bounds, respectively. Then,

$$\Delta t_i^\nu = \begin{cases} (C_i^\nu - 1)t_i^\nu = (\alpha^\nu - 1)\left[1 + \frac{1}{a_i}\sum_{j=1}^{N} \lambda_j^\nu \left(\frac{\partial c_j}{\partial t_i}\right)^\nu\right] t_i^\nu, & i \varepsilon A_v^\nu \\ (t_i)_{min} - t_i^\nu \quad , & i \varepsilon P_\ell^\nu \\ (t_i)_{max} - t_i^\nu \quad , & i \varepsilon P_u^\nu \end{cases} \tag{17}$$

To first order,

$$(\Delta c_j')^\nu = \Delta c_j^\nu + 2z_j^\nu \Delta z_j^\nu \tag{18}$$

and

$$\Delta c_j^\nu = \sum_i \left(\frac{\partial c_j}{\partial t_i}\right)^\nu \Delta t_i \quad , \quad i\varepsilon\left(A_v^\nu + P_\ell^\nu + P_u^\nu\right) \tag{19}$$

After use of equations (17) for Δt_i^ν, equation (19) becomes

$$\Delta c_j^\nu = (\alpha^\nu - 1)\left(\beta_j^\nu + \sum_{k=1}^{N} \beta_{kj}^\nu \lambda_k^\nu\right) + \sum_{i\varepsilon P_\ell^\nu} \left(\frac{\partial c_j}{\partial t_i}\right)^\nu \left[(t_i)_{min} - t_i^\nu\right]$$

$$+ \sum_{i\varepsilon P_u^\nu} \left(\frac{\partial c_j}{\partial t_i}\right)^\nu \left[(t_i)_{max} - t_i^\nu\right] \tag{20}$$

where

$$\beta_j^\nu = \sum_{i\varepsilon A_v^\nu} \left(\frac{\partial c_j}{\partial t_i}\right)^\nu t_i^\nu \tag{21}$$

$$\beta_{kj}^\nu = \sum_{i\varepsilon A_v^\nu} \frac{1}{a_i} \left(\frac{\partial c_k}{\partial t_i}\right)^\nu \left(\frac{\partial c_j}{\partial t_i}\right)^\nu t_i^\nu \tag{22}$$

Equations (14) and (16) yield

$$\Delta z_j^{\nu} = (\alpha^{\nu}-1)K\lambda_j^{\nu}z_j^{\nu} \tag{23}$$

Equations (20) and (23) can now be combined to give an expression for $(\Delta c_j')^{\nu}$. Setting this quantity to zero for all j, letting $(z_j^2)^{\nu} = -c_j^{\nu}$, and rearranging the resulting equations gives a linear set of equations for the Lagrange multipliers λ_j^{ν}:

$$\sum_{k=1}^{N} \bar{\beta}_{kj}^{\nu} \lambda_k^{\nu} = - \beta_j^{\nu} - \left\{ \sum_{i\varepsilon P_\ell^{\nu}} \left(\frac{\partial c_j}{\partial t_i}\right)^{\nu} \left[(t_i)_{min}-t_i^{\nu}\right] + \sum_{i\varepsilon P_u^{\nu}} \left(\frac{\partial c_j}{\partial t_i}\right)^{\nu} \left[(t_i)_{max}-t_i^{\nu}\right]\right\} / (\alpha^{\nu}-1) \tag{24}$$

where

$$\bar{\beta}_{kj}^{\nu} = \beta_{kj}^{\nu} - 2Kc_j^{\nu}\delta_{kj} \tag{25}$$

and δ_{kj} is the Kronecker delta function.

Choosing the step-size parameter $(\alpha^{\nu}-1)$ completes this algorithm. There are four criteria that are applied; these follow, with slight modifications, the criteria set forth in reference 4. The criteria are as follows:

(i) Limit on step size.

The change in each design variable is given by

$$\Delta t_i^{\nu} = (\alpha^{\nu}-1)v_i^{\nu}t_i^{\nu}, \tag{26}$$

where $v_i = \bar{v}_i/a_i$. The magnitude of the vector whose components are $\Delta t_i^\nu/t_i^\nu$ is

$$|\overrightarrow{\Delta t_i^\nu/t_i^\nu}| = |\alpha^\nu-1| \left[\sum_{i\varepsilon A_\nu} \left(v_i^2\right)^\nu\right]^{1/2} \tag{27}$$

If δ_t^2 is now defined as the maximum permissible magnitude of $\Delta t_i^\nu/t_i^\nu$, then the left-hand side of equation (27) is $\delta_t(N_A)^{1/2}$, where N_A is the number of design variables in the active set. Equation (27) can then be rewritten as a limiting criterion on $|\alpha^\nu-1|$:

$$|\alpha^\nu-1| \leq \delta_t(N_A)^{1/2}/\left[\sum_{i\varepsilon A_\nu} \left(v_i^2\right)^\nu\right]^{1/2} \tag{28}$$

(ii) Feasibility of the new design.

This criterion is applied in order to ensure that a design that is near a constraint boundary $c_j = 0$ will not be driven too far from it. Equations (12), (18), and (23) can be combined for $(\Delta c_j')^\nu = 0$ to give to first order

$$\Delta c_j^\nu = 2(\alpha^\nu-1)K\lambda_j^\nu c_j^\nu \tag{29}$$

With δ_c defined as the maximum allowable value of $|\Delta c_j^\nu/c_j^\nu|$, equation (29) becomes

$$|\alpha^\nu-1| \leq \delta_c/2K|\lambda_j^\nu| \tag{30}$$

(iii) Bounds on the design variables.

This criterion is used to ensure that no design variable is driven too far outside its allowable range. Tolerance bands on the upper and lower bounds are established with parameters $t_u(>1)$ and $t_\ell(<1)$, respectively. With the proviso that $(\alpha^\nu-1)$ be always negative, this criterion takes two forms. From equation (26), it can be seen that the upper or lower bound applies if v_i^ν is respectively negative or positive. Hence, the criterion is one of the following:

$$|\alpha^\nu-1| \leq \left[t_i^\nu-t_\ell(t_i)_{min}\right]/v_i^\nu t_i^\nu \ , \ v_i^\nu > 0 \tag{31}$$

$$|\alpha^\nu-1| \leq \left[t_i^\nu-t_u(t_i)_{max}\right]/v_i^\nu t_i^\nu \ , \ v_i^\nu < 0 \tag{32}$$

(iv) Convergence

As the design nears an optimum, it is possible that the criteria outlined above will all provide relatively large upper bounds on $|\alpha^\nu-1|$. To ensure that the algorithm remains stable, this criterion merely establishes an upper limit on $|\alpha^\nu-1|$:

$$|\alpha^\nu-1| \leq d \tag{33}$$

At any step in the iterative process, the value of $(\alpha^\nu-1)$ to be used is given by

$$(\alpha^\nu-1) = -|\alpha^\nu-1|_{min} \tag{34}$$

where the right side is given by the minimum of the upper bounds computed from equations (28), (30), (31), (32), or (33).

There are additional details that need to be discussed concerning the assignment of a design variable to the passive set. The parameters t_ℓ and t_u are used to define tolerance bands on both upper and lower bounds. In other words, a design variable is consigned to the passive set and given the appropriate upper- or lower-bound value whenever

$$t_i^{\nu+1}/(t_i)_{min} \leq t_u \tag{36}$$

or

$$t_i^{\nu+1}/(t_i)_{max} \geq t_\ell \qquad (37)$$

It is intended here to accelerate as much as possible the selection of active and passive variables. When t_u and t_ℓ are not equal to unity, this means that a design variable may be selected as passive, even though $(\alpha^\nu-1)$ has not been selected from equations (31 or (32). As a consequence of this, either of the sets P_ℓ^ν or P_u^ν will no longer be null, and it will be necessary to return to equation (24) and repeat the calculation of the Lagrange multipliers. In principle, this process must be repeated until the selection of the passive variables converges; in practice, only one repetition is usually necessary. Note also that if t_ℓ and t_u are set equal to unity, then the choice of $(\alpha^\nu-1)$ will be governed by one of the equations (31) or (32) and no repetition will be needed, since there is no inconsistency in equations (17), which give the increments Δt_i^ν.

Since all of the a_i's are positive, convergence to a local minimum is measured by the v_i, and the convergence criterion is

$$|v_i^\nu| \leq \varepsilon_v \ , \quad i\varepsilon A_v^\nu \qquad (38)$$

where ε_v is a user-supplied quantity. To ensure that no passive design variables need to be reintroduced into the active set, the applicable Kuhn-Tucker conditions, equations (5) and (6), are verified (with v_i^ν rather than $\bar{v}_i^\nu$):

$$v_i^\nu \geq 0 \ , \quad i\varepsilon \bar{P}_\ell^\nu \qquad (39)$$

$$v_i^\nu \leq 0 \ , \quad i\varepsilon \bar{P}_u^\nu \qquad (40)$$

Here $\bar{P}_{\ell}^{\nu}$ and $\bar{P}_{u}^{\nu}$ define the complete set of passive design variables. Since the redesign formula gives equation (26) for Δt_i^{ν} , there is an obvious physical explanation for the above conditions. If a design variable t_i is at its minimum value and its associated convergence function v_i is negative, a redesign will produce a positive increment Δt_i, thereby implying that this design variable should be reintroduced into the active set. Similar reasoning holds for a design variable at its upper bound when its convergence function is positive.

3.2 Summary

The algorithm is summarized in the following steps:

(1) For the current design, evaluate the constraints c_j, $j=1,2,\ldots,N$.

(2) Calculate the derivatives $\partial c_j/\partial t_i$, $i=1,2,\ldots,M$; $j=1,2,\ldots,N$.

(3) Calculate β_j from equation (21) and $\bar{\beta}_{kj}$ from equations (22) and (25).

(4) Solve for the λ_k from equation (24), with P_{ℓ} and P_u null if this is the first time the multipliers are calculated for this iteration. If not, include terms for the appropriate design variables identified in step (8) below.

(5) Compute the convergence functions $v_i = \bar{v}_i/a_i$; $\bar{v}_i$ is given by equation (4).

(6) Test for convergence, using equation (38). If this test is satisfied, test the passive variables with equations (39) and (40). If any of these tests fails, redefine the active and passive sets appropriately and return to step (3). If the tests are satisfied, exit with the final design information.

(7) Compute $(\alpha^{\nu}-1)$ from equations (28) and (30)-(34).

(8) Compute the new design variables according to the first of equations (13) and equation (15). Test for new passive variables using equations (36) or (37). If necessary, redefine the active and passive sets and return to step (3). Otherwise, compute the new mass and return to step (1) to begin a new iteration.

In brief, this algorithm seeks to avoid identifying active constraints by using slack variables to permit all of the constraints to be treated as equality constraints. Implied in this approach is the assumption that all intermediate designs and the starting design must be feasible - that is, no constraint violations are tolerated. Also, all constraints are to be evaluated at every design, as the algorithm is currently formulated. For truly large systems, where a great number of constraints may be applied, it would be necessary to introduce a "throwaway" concept or some similar strategem, whereby constraints that are nowhere near being active are at least temporarily ignored.

4. AN OPTIMALITY-CRITERION METHOD EMPLOYING GAUSS-SEIDEL ITERATION

4.1 General Description

This method is based on the work of reference 5. The strategy here is to let the active constraints be selected by the signs of the Lagrange multipliers. The Gauss-Seidel iterative procedure is used to permit the constraints associated with negative multipliers to be eliminated efficiently. At the optimum, the only constraints involved are the active ones, for which the Lagrange multipliers are non-negative.

The problem statement is given by equations (1)-(8), with the redesign algorithm given by equations (13) and (15). In order to reduce the number of constraints included in the calculation of the λ_j, a positive tolerance parameter $\bar{\delta}_c^\nu$ is specified, and those constraints are selected for which

$$c_j^\nu \geq - \bar{\delta}_c^\nu \tag{41}$$

Let A_c^ν define the set of constraints for which this test is satisfied. These constraints are either potentially active or violated.

Computation of the first-order approximation for Δc_j^ν gives equation (20). If it is now assumed that all the constraints in A_c^ν are to be active constraints, then it is required that $\Delta c_j^\nu = -c_j^\nu$, and equation (20) can be manipulated to give

$$\sum_{k\varepsilon A_c^\nu} \beta_{kj}^\nu \lambda_k^\nu = -\beta_j^\nu - \left\{ c_j^\nu + \sum_{i\varepsilon P_\ell^\nu} \left(\frac{\partial c_j}{\partial t_i}\right)^\nu \left[(t_i)_{min} - t_i^\nu\right] + \sum_{i\varepsilon P_u^\nu} \left(\frac{\partial c_j}{\partial t_i}\right)^\nu \left[(t_i)_{max} - t_i^\nu\right] \right\} / (\alpha^\nu - 1), \; j\varepsilon A_c^\nu \qquad (42)$$

The coefficients β_j^ν and β_{kj}^ν are given by equations (21) and (22) respectively, with $j,k\varepsilon A_c^\nu$.

The next step is to determine that subset of A_c^ν for which solutions of equations (42) will produce non-negative values of λ_k^ν. One approach is to solve equations (42) for the set of A_c^ν, eliminate the terms associated with any negative multipliers, and repeat this process until only non-negative values of the multipliers are computed. However, as noted in reference 8, there is no guarantee that this procedure will converge, because of the coupling among the equations provided by the coefficients β_{kj}^ν , $k \neq j$. The solution proposed in reference (5) is to apply the Gauss-Seidel iterative procedure to equations (42). Let these equations be rewritten in matrix form as

$$[\bar{B}]\{\lambda\} = \{\bar{b}\} \qquad (43)$$

and let n be their order. Then, let equation (43) be recast in the following form:

$$\{\lambda\} = [B]\{\lambda\} + \{b\} \qquad (44)$$

where

$$B_{ij} = \bar{B}_{ij}/\bar{B}_{ii} - 1 \ , \quad i,j = 1,2,\dots,n \qquad (45)$$

$$b_i = b_i/B_{ii} \qquad , \quad i = 1,2,\dots,n \qquad (46)$$

Thus, [B] is a matrix with a zero principal diagonal, and the Gauss-Seidel iterative algorithm is

$$\lambda_i^{\mu} = \sum_{j=1}^{i-1} B_{ij}\lambda_j^{\mu} + \sum_{j=i+1}^{n} B_{ij}\lambda_j^{\mu-1} + b_i \ , \quad i = 1,2,\dots,n \qquad (47)$$

Here μ is the Gauss-Seidel iteration number, and the terms in the first summation for $i = 1$ and the second summation of $i = n$ are ignored. As this algorithm is applied, each multiplier λ_k is estimated from current values for $i < k$. This means that whenever a negative value for a multiplier is determined, it can be immediately deleted, with minimal impact on the values computed for the remaining multipliers in that particular iteration. Although it cannot be rigorously proven, this process is expected to lead naturally to the proper subset of A_c^{ν} (ref. 5). Let this subset be $\bar{A}_c^{\nu}$.

Redesign can now be carried out with equations (13) and (15), with $j \epsilon \bar{A}_c^{\nu}$ in equation (15). The process of identifying passive design variables is similar to that described in Section 3 above. If any design variables are predicted to reach their upper or lower bounds, then the calculations of the multipliers must be repeated, with the appropriate terms added to equation (42). Unlike the original algorithm proposed in reference 5, design variables remain passive until a converged design is found. At that point the tests given by equations (39) and (40) are applied, and the appropriate variables reintroduced into the active set if any of the tests fails.

The parameters α^{ν} and δ^{ν} are scheduled as follows:

$$\alpha^{\nu} = \alpha_x \alpha^{\nu-1} \tag{48}$$

$$\bar{\delta}_c^{\nu} = \delta_x \bar{\delta}_c^{\nu-1} \tag{49}$$

To accelerate convergence, α_x can be chosen as a positive number greater than unity, with $\alpha^{(0)}$ given as a negative number. The multiplier δ_x is typically less than unity, since it is desirable to have equation (41) provide an increasingly restrictive test as the optimization proceeds.

Convergence is measured by two criteria:

$$|c_j^{\nu}| < \varepsilon_c \quad , \quad j \varepsilon \bar{A}_c^{\nu} \tag{50}$$

$$|v_i^{\nu}| < \varepsilon_v \quad , \quad i \varepsilon A_v^{\nu} \tag{51}$$

4.2 Summary

This algorithm is summarized as follows:

(1) For the current design, evaluate the constraints c_j, $j = 1,2,\ldots,N$.

(2) Calculate the derivatives $\partial c_j/\partial t_i$, $i = 1,2,\ldots,M$, $J = 1,2,\ldots,N$.

(3) Use equation (41) to test for the set A_c^{ν} of potentially active or violated constraints.

(4) Compute β_j^{ν} and β_{kj}^{ν} from equations (21) and (22), respectively, with $j,k \varepsilon A_c^{\nu}$.

(5) Use the Gauss-Seidel iterative formula, equation (47) to determine the constraint set $\bar{A}_c^{\nu}$.

(6) Compute the redesign factors C_i^{ν} from equation (15) and identify

any new passive variables, redefine the active and passive sets accordingly and return to step (4), with the appropriate terms added in equation (42). Continue this loop through steps (4)-(6) until there is no change in the active-passive identities.

(7) Compute the *convergence functions* v_i^{ν} *for the constraint* set $\bar{A}_c^{\nu}$ and test for convergence with equations (50) and (51). If these tests are satisfied, test the passive variables with equations (39) and (40). If any of these tests fails, redefine the active and passive sets accordingly and return to step (4). If not, exit with the final design information.

(8) If any of the convergence tests fails, compute the new design variables (equations (13)) and the new values of the step-size parameter α and the constraint-tolerance parameter $\bar{\delta}_c$ (equations (48) and (49)) and return to step (1) to begin a new iteration.

It is worth noting that this algorithm also requires that all constraints be evaluated at each iteration. The selection of the active constraints - that is, those that are actually included in the redesign - is a two-step process. First, the constraints that are violated or are within a given tolerance band around zero are chosen, and from these the active set is chosen by eliminating those for which the multipliers are negative. It is therefore to be expected that this strategy would be most valuable when there is a large number of constraints, and the active set is rapidly changing from iteration to iteration. When the constraints are so numerous that it is not feasible to evaluate them all at every iteration, a "throwaway" concept would be useful and could easily be incorporated.

5. *CONSTRAINT EVALUATIONS AND DERIVATIVE CALCULATIONS*

Three types of constraints are considered - displacement in response to a fixed static load, natural frequencies, and flutter speed. The evaluations of the constraints and their derivatives follow generally the procedures outlined in reference 1. The basic formulas will be repeated below for the sake of completeness and any differences between current usage and that in reference 1 will be described.

5.1 Displacement Constraint

The constraint is given by

$$c = \frac{u_r}{(u_r)_{des}} - 1 \qquad (52)$$

and the derivative is calculated by the dummy-load method:

$$\frac{\partial c}{\partial t_i} = \frac{1}{(u_r)_{des}} \frac{\partial u_r}{\partial t_i} = - \frac{1}{(u_r)_{des}} \lfloor u \rfloor [K_i] \{u^{(r)}\} \qquad (53)$$

Here {u} is the set of nodal displacements associated with a given loading condition, whose rth component u_r is constrained to be less than or equal to $(u_r)_{des}$, and $\{u^{(r)}\}$ is the set of displacements resulting from the application of a dummy load at the rth node in the direction of u_r. The matrix $[K_i]$ is the stiffness matrix associated with design variable t_i; both mass and stiffness matrices vary linearly with the design variables. For multiple constraints, different loading conditions, and different displacements for each loading condition, can be considered.

5.2 Frequency Constraints

In contrast to the expression in reference 1, the constraint function for a natural-frequency constraint is written as

$$c = 1 - \frac{\omega_r}{(\omega_r)_{des}} \qquad (54)$$

where ω is the rth free-vibration frequency of the structure, and it is constrained to be greater than or equal to $(\omega_r)_{des}$. The derivative expression is

$$\frac{\partial c}{\partial t_i} = - \frac{1}{(\omega_r)_{des}} \frac{\partial \omega_r}{\partial t_i} \tag{55}$$

where

$$\frac{\partial \omega_r}{\partial t_i} = \frac{\lfloor q^{(r)} \rfloor [GK_i] \{q^{(r)}\} - \omega_r^2 \lfloor q^{(r)} \rfloor [GM_i] \{q^{(r)}\}}{2\omega_r \lfloor q^{(r)} \rfloor [GM] \{q^{(r)}\}} \tag{54}$$

It is assumed here that the natural modes of the initial design are used to define generalized coordinates for all the designs, so that the generalized derivative matrices $[GK_i]$ and $[GM_i]$ can be written as an inactive part plus the sum of the active parts. The column $\{q^{(r)}\}$ is the rth mode shape of the *current* design (see reference 1). Multiple frequency constraints can be considered.

5.3 Flutter-Speed Constraints

Here, the constraint function c is identified directly with the damping parameter g. In essence, this formulation requires that the imaginary part of a critical root of the flutter equations be non-positive at a given free-stream speed and altitude. The mode shapes of the initial design are used to define generalized coordinates, so the generalized derivative matrices $[GK_i]$ and $[GM_i]$ are constant, and the generalized aerodynamic matrix [GA] varies only with reduced frequency k. The derivative is evaluated just as in reference 1:

$$\frac{\partial c}{\partial t_i} = \frac{\partial g}{\partial t_i} = \Bigg[\left(R_2^i - \omega^2 R_1^i - gI_2^i\right)\left(2gR_3 + 2I_3 + \frac{b\omega^3}{V_\infty} I_4\right) - \left(2R_3 - 2gI_3 + \frac{b\omega^3}{V_\infty} R_4\right)\left(I_2^i - \omega^2 I_1^i + gR_2^i\right)\Bigg]/D \tag{57}$$

where

$$D = \left(2gR_3 + 2I_3 + \frac{b\omega^3}{V_\infty} I_4\right)I_3 + \left(2R_3 - 2gI_3 + \frac{b\omega^3}{V_\infty} R_4\right)R_3 \tag{58}$$

$$R_1^i = \mathrm{Re}\left(\lfloor \bar{p} \rfloor [GM_i] \{\bar{q}\}\right) \tag{59}$$

$$R_2^i = \mathrm{Re}\left(\lfloor \bar{p} \rfloor [GK_i] \{\bar{q}\}\right) \tag{60}$$

$$R_3 = \mathrm{Re}\left(\lfloor \bar{p} \rfloor [GK] \{\bar{q}\}\right) \tag{61}$$

$$R_4 = \mathrm{Re}\left(\lfloor p \rfloor \left[\frac{\partial GA}{\partial k}\right] \{\bar{q}\}\right) \tag{62}$$

and I_1^i, I_2^i, I_3, and I_4 are corresponding imaginary parts. Also, V_∞ is the free-stream speed, b is a reference length, ω is the frequency, $\{\bar{q}\}$ is the eigenvector for the critical flutter mode, and $\{\bar{p}\}$ is the adjoint eigenvector.

When the constraint is evaluated, it is necessary to ensure that the reduced frequency, which is fixed in order to evaluate [GA], is compatible with the frequency computed from the eigenvalue $\bar{\Omega} = (1+ig)/\omega^2$ (here $i = (-1)^{1/2}$). While it is possible to compute $\frac{\partial \omega}{\partial t_i}$ and use it to estimate the new value of the frequency once the design change has been compuated, it is simpler to iterate. The previous value of reduced frequency is used to define the aerodynamic matrix [GA], and a value of ω computed from the eigenvalue $\overline{\Omega}$. If this value is not equal to within some specified tolerance to the value used to compute the reduced frequency, it is used as the new reduced frequency and the matrix equation for flutter is solved again. Typically only a few iterations are needed; more details can be found in reference 1.

Multiple flutter-speed constraints may be considered. These may be critical speeds at different altitudes or Mach numbers, or they may involve additional potentially critical branches for a single case.

5.4 Accuracy of Derivative Calculations

In order to test the accuracy of the derivative calculations, the analytical derivatives, computed as described above for frequency and flutter constraints, were compared with derivatives computed by finite differencing. The structure used was the sandwich wing of reference 9. This wing had a primary flutter constraint corresponding to $M_\infty = 0.8$ at an altitude of 2,438 m, a secondary flutter constraint corresponding to the crossover of another root in the V-g plane at a slightly higher airspeed, and a frequency constraint that required the fundamental natural frequency to be not less than that of the initial design. Additional details concerning this wing and the constraints are given in Section 6.

Finite-difference derivatives with respect to all seven design variables are compared to analytical derivatives in Table 1. These derivatives were computed at an intermediate design obtained from an optimization run, so the modes used to define generalized coordinates are primitive modes. As can be seen, the derivatives calculated by the two methods are generally in very good agreement. Finite-difference derivatives were computed for a number of different perturbation magnitudes and compared in order to minimize numerical errors; the derivatives given in the tables were determined with 5% perturbations.

During this portion of the study, it was discovered that the convergence test used in the iterating to determine the new frequency when the flutter constraint was being evaluated had to be tightened considerably in order to ensure sufficiently accurate determination of the damping parameter g. It was ultimately necessary to require that the selected and computed values of reduced frequency agree to within 0.0001.

As noted in reference 10 and elsewhere, the use of fixed modes does lead to inaccuracies both in constraint evaluation and in derivative calculation. On the other hand, updating the modes after every iteration, or even after a certain number of iterations, is often not feasible because of the additional computer time required or the abrupt changes introduced in constraints and derivatives when the modes are changed. In principle, using a sufficiently large number of modes, even if they are used as fixed modes, should resolve this difficulty. However, there are practical limits to the number of modes that can be used. For example, mode shapes are often fit with polynomials or splines before they are used in calculating generalized aerodynamic forces, and the accuracy of such fits for high-order modes with complicated node-line patterns is questionable.

Another approach has been proposed in reference 5. It was hypothesized that the principal source of modeling errors with fixed modes is the failure of the inplane portions of the modes to produce negligible inplane forces as the design is changed. To avoid this problem, it was proposed to preserve unchanged the transverse displacements but recompute the inplane displacements from the requirement that the inplane forces be zero. This would involve additional computations in constructing the generalized mass and stiffness matrices but would not affect the generalized aerodynamic forces. In reference 5, the inplane displacements were recomputed, but the additional terms that arise in the derivative calculations to account for changes in certain portions of the modes were ignored. The accuracy of constraint evaluation was enhanced by this strategem, and it is expected that accounting for the mode-shape changes in the derivative calculations, which involves only the derivatives of the generalized mass and stiffness matrices, should improve the accuracy of the derivative calculations with little added computational effort.

6. NUMERICAL EXAMPLES

6.1 Test Cases

It is instructive to consider first some test cases involving two design variables, so that the progress of the optimization can be plotted and visualized in design space. For all of the test cases, the merit function was

$$M = 2t_1 + 3t_2 \tag{63}$$

For the first test case, the "behavioral" constraints were

$$c_1 = t_2 - t_1^2 \leq 0 \tag{64}$$

$$c_2 = 1 - \left(t_1^2 + t_2^2\right)/2 \leq 0 \tag{65}$$

and minimum-gage constraints of $(t_1)_{min} = 0.2$, $(t_2)_{min} = 0.5$ were also imposed. Figure 1 presents the path in design space taken by the slack-variable program (program SLACK) from an initial design at (2.0,1.5). As can be seen the first few steps follow reasonably closely a steepest-descent path, nearly normal to the $M = \text{const.}$ lines, until the minimum-gage constraint on t_2 is encountered. Thereafter the designs proceed along this constraint boundary until the intersection with the $c_2 = 0$ boundary is reached, at (1.3228, 0.5000). The value of the merit function at this point, which also is a global minimum, is $(M)_{min} = 4.1456$. After 42 iterations, the computed values for the optimum were $(M)_{min} = 4.1460$ at (1.3230, 0.5000). Values of other parameters used in SLACK are given in the caption. No attempt was made at this point to vary these parameters so as to reduce the number of iterations required.

The next set of slack test cases was run with the sign on c_1 changed, so that the feasible domain was to the left of the $c_1 = 0$ boundary. With the same minimum-gage constraints, there are multiple extrema here - one at (0.2000, 1.4000), with $(M)_{min} = 4.600$, and one at (1.0000, 1.0000), with $(M)_{min} = 5.000$. With the initial design at (0.9000, 1.2000), the latter extremum is found after 44 iterations (figure 2); the computed values are $(M)_{min} = 5.0019$ at (0.9981, 1.0020). If the initial design is moved to (1.2000, 2.000), the other extremum is found after 64 iterations (figure 3). The computed values here are $(M)_{min} = 4.6006$ at (0.2000, 1.4002). In figures 2 and 3, a line is drawn through the origin in the steepest-descent direction. The intersection of this line with the $c_2 = 0$ boundary (point A) defines the maximum value of M on the boundary. It is therefore readily apparent that any optimization step reaching the $c_2 = 0$ boundary to the right of point A will result in an optimum solution to the right, and in similar fashion the optimum to the left will be found for any step reaching the $c_2 = 0$ boundary to the left.

Another test case with the same constraints c_1 and c_2, but with $(t_1)_{min} = 0.500$, $(t_2)_{min} = 1.500$, was run to test the operation of SLACK when the optimum design is dictated by the minimum-gage constraints. This optimum was found after 17 iterations (figure 4).

The final SLACK test case involves two concave constraints - the same constraint function c_1 used above, and

$$c_2 = (t_1-2)^2 + (t_2-2)^2 - 2.56 \leq 0 \qquad (66)$$

Minimum-gage constraints of (0.4000, 0.4000) were imposed, but these were inactive, and the global optimum is at (0.9102, 0.8285), with $(M)_{min} = 4.3060$ (figure 5). With an initial design of (1.000, 2.000), the optimum was found after 32 iterations. The computed values were $(M)_{min} = 4.3084$ at (0.9072, 0.8313).

It is interesting to note that the behavior of the slack-variable algorithm, as given by the paths in design space plotted for these test cases, closely follows a gradient-projection strategy: steepest descent until a constraint surface is encountered, and then steps along the projection of the gradient of the merit function on the tangent to the constraint surface (see, for example, reference 11). This is not so surprising, in view of the close relationship between a Kiusalaas-type algorithm and a projection algorithm (reference 12).

A number of cases with different parameters in SLACK were run with the constraints and initial design given in figure 5. The results are summarized in Table 2. As would be expected, imposing a more severe convergence criterion results in an increased number of iterations required - in this case, the number of iterations is almost doubled for a factor of 10 reduction in ε_v. Comparing numbers of iterations with different values of K and δ_c suggests that larger values of δ_c and smaller values of K will enhance convergence, at least in cases where many iterations take place along a nonlinear constraint boundary. This is in fact expected to occur for most problems of practical significance.

One test case was run with the program based on Gauss-Seidel iteration - program GAUSS. This case involves the same merit function as previously used and three constraints:

$$c_1 = t_1^2 - t_2 \leq 0 \qquad (67)$$

$$c_2 = -1 + \left(t_1^2 + t_2^2\right)/2 \leq 0 \qquad (68)$$

$$c_3 = 2\left(1-t_1^2\right) - t_2 \leq 0 \qquad (69)$$

These constraint boundaries and the results from a run with program GAUSS are presented in Figure 6. The feasible domain is a closed area roughly triangular in shape, with a global optimum at the intersection of the boundaries of the two active constraints, c_1 and c_3. The optimum is at (0.8165, 0.6667), with $(M)_{min} = 3.6330$. With an initial design at (0.8200, 0.9300), the optimal solution was found in just four interations to be $(M)_{min} = 3.6718$ at (0.8115, 0.6830). As can be seen in the figure, there were some oscillations about the optimum. These were undoubtedly the result of the values used for the various parameters, which are given in the caption. Note in particular that a tighter convergence test would have located the computed minimum closer to the exact one. No studies of the effects of variations in these parameters were performed at this time, since it was decided to concentrate the remaining efforts on more complicated problems with program SLACK.

6.2 Biconvex Sandwich Wing

The structure chosen for the remaining examples was the biconvex sandwich wing of reference 9, example 2. A planform view of this wing is given in figure 7, which shows the node-point and cover-panel numbering. The coordinates of the nodes are given in reference 9. To model the core, thick shear webs were placed along each chordwise and spanwise line between node points, and axial elements were placed between upper-surface and lower-surface node points, except at the wing root. These elements were used to contribute stiffness only. Triangular cover panels, numbered as indicated in figure 7 for the upper surface, were used to model the skin. The complete model had 76 nodes, 14 of which were restrained at the root; 102 cover elements; 49 shear webs; and 31 axial elements. There were 186 unrestrained degrees of freedom, which were collapsed to 31 transverse degrees of freedom at the upper surface. The design variables were taken to be the thicknesses of the cover panels. Each design variable governed the thickness of an upper cover panel and its lower-surface counterpart, so there were a total of 51 design variables, which could be additionally linked in any combination to create smaller problems. Concentrated tuning masses were also located at nodes 18 and 56; these were controlled by a single additional concentrated-mass design variable. Finally, nonactive concentrated masses were located at all the nodes to represent internal fuel. The values used for these masses were obtained by subtracting the active mass contributed by the cover panels from the total node-point masses given in reference 9.

Material properties for the cover and web elements were the same as those given in reference 9: $E = 1.1308 \times 10^8$ kPa, $\rho = 4.4244 \times 10^{-6}$ kg/cm^3, and $\bar{\nu} = 0.3$ for the covers; and $G = 4.3492 \times 10^7$ kPa and $\bar{\nu} = 0.3$ for the webs. The axial elements, which had no counterparts in the model of reference 9, had $EA = 2.1440 \times 10^4$ kPa-m^2. These elements were needed in the present model to stabilize the shear webs, which were different from the webs used in reference 9. It should be noted that the webs and axial elements were used to provide an extremely stiff, weightless core. As long as these elements are stiff enough, the transverse mode shapes and frequencies of the wing will not be very sensitive to the values used for these stiffnesses. The webs were all assumed to be 9.8 in. thick, and the cover panels were sized according to the initial design of example 2 in reference 9 - 0.2540 cm for panels 1-43 and 52-94, and 0.05080 cm for panels 44-51 and 95-102. The initial weight for this wing was 67,000 kg, including the fuel, with 7,602 kg available for optimization This includes 350.3 kg for the tuning masses.

In the examples that follow, the design variables were linked as indicated in figure 8, so that the number of design variables was reduced to seven, including the concentrated-mass design variable (no. 7).

6.3 Optimization with Frequency Constraints

Mode shapes and frequencies were computed for the wing described above, and the wing was then optimized by program SLACK with the requirement that the first two natural frequencies be greater than or equal to 6.20 Hz and 18.70 Hz, respectively. The corresponding frequencies of the initial design were slightly in excess of these values. The first 12 modes of the initial design were used as fixed modes to define generalized coordinates. After 34 iterations, an optimum weight (less fuel) of 569.9 kg was computed. An iteration history of the variable weight is presented in figure 9, and the optimal values of the design variables are presented in figure 10. Around 25% of the weight available for optimization was removed. Both frequency constraints were active at the optimum, as would be expected. CPU time required on an IBM 370/78 computer was approximately 0.1 sec per iteration. Parameters used in SLACK were $\delta_t = \delta_c = 0.5$, $t_\ell = 0.9$, $d = 2.5$, $K = 0.1$, and $\varepsilon_v = 0.001$. Minimum values of 0.02540 cm were specified for design variables 1-5, while design variables 6 and 7 had minima of 0.01016 cm and 35.03 kg, respectively.

The iteration history of figure 9 demonstrates the usual rapid reduction in weight, followed by relatively small weight reductions as the optimum solution is approached.

6.4 Optimization with Flutter Constraints

The same wing was next analyzed for flutter. The initial design, with 12 modes again used, had a critical flutter speed of 264 m/sec at an altitude of 2,438 m, which was a matched flutter point for $M_\infty = 0.8$. Doublet-lattice aerodynamic generalized forces were computed with a code based on that used in reference 13. Polynomial approximations to the mode shapes were used, and computed aerodynamic loads at discrete values of reduced frequency were fitted with polynomials to provide input for the flutter-analysis code and later on for program SLACK, where derivatives of these loads with respect to reduced frequency were required.

Optimization was originally attempted with a single flutter constraint, $g \leq 0$ at 262 m/sec. However, after six iterations, another root became critical at a much lower speed, and it was necessary to constrain this as well. The branches of interest are plotted in V-g space in figure 11 for the design obtained after the fifth iteration. The mode 2 crossover at 265 m/sec is the one originally constrained. To prevent the mode 4 crossover from moving to a lower speed than this, a second constraint was imposed, namely $g \leq 0$ at 276 m/sec, as shown in figure 11. This configuration was then optimized, and the results are given in figures 12 and 13. The parameter values used in SLACK were the same as those used for the optimization with multiple frequency constraints, and the same minimum values for the design variables were also imposed.

In figure 12, the initial weight is that at the end of the fifth iteration from the initial design with a single flutter constraint, which is 6,338 kg. The optimum, found after another 40 iterations, is 5,647 kg, which is 74% of the true initial weight, less fuel, of 7,602 kg. The optimal skin thickness distribution shown in figure 13 is quite different from that found for multiple frequency constraints. Approximately 1.2 sec per iteration were required by program SLACK for this problem.

6.5 Optimization with Flutter and Frequency Constraints

A single flutter constraint ($g \leq 0$ at 262 m/sec) and a single frequency constraint ($\omega_1 \geq 6.20$ Hz) were then imposed. After five iterations, a mode 4 crossover at a speed less than 262 m/sec was again found. A flutter analysis of the fourth-iteration design produced mode 2 and mode 4 crossovers similar to those in figure 11, except that the mode 4 crossover was at 305 m/sec. Another flutter constraint was added - $g \leq 0$ at 305 m/sec - and an optimal solution was found after 17 more iterations. The results for this problem are given in figures 14 and 15. The initial weight in figure 14 was that of the fourth-iteration design - 6,466 kg. The final weight was virtually identical to that found with the flutter constraints alone - 5,653 kg. This can also be seen in figure 15, where the optimal skin thickness distribution is very similar to that given in figure 13. The previous parameter values were used here in SLACK, and approximately 1.5 sec per iteration were required.

7. CONCLUDING REMARKS

As noted in the discussion of the Gauss-Seidel iterative procedure for determining the Lagrange multipliers, the strategems employed in program GAUSS are most effective when there are a large number of constraints, so that determining the set of constraints that is active at the optimum is a genuine problem. In particular, this should be the case when stress constraints are added to the behavioral constraints discussed in this report. Adding the capability to compute stresses and their derivatives is therefore a prime requirement in order to test program GAUSS adequately.

The results from the optimization problems treated with program SLACK suggest that this program is reliable and easy to use. Convergence properties, however, could be enhanced, probably by a better selection of the various parameters that need to be supplied by the user. More studies with variations of these parameters would be useful. Also, it is time to treat larger problems, and particularly to consider problems with stress constraints, so that comparisons with program GAUSS can be undertaken.

REFERENCES

1. McIntosh, S. C., Jr.: Modifications and Improvements in a Structural Optimization Scheme Based on an Optimality Criterion. NEAR TR 169, June 1978.

2. Segenreich, S. A. and McIntosh, S. C., Jr.: Weight Minimization of Structures for Fixed Flutter Speed via an Optimality Criterion. AIAA Paper No. 75-779, presented at AIAA/ASME/SAE 16th Structures, Structural Dynamics, and Materials Conference, Denver, CO, May 27-29, 1975.

3. Segenreich, S. A. and McIntosh, S. C., Jr.: Weight Optimization Under Multiple Equality Constraints Using an Optimality Criterion. Paper presented at AIAA/ASME/SAE 17th Structures, Structural Dynamics, and Materials Conference, King of Prussia, PA, May 5-7, 1976.

4. Segenreich, S. A., Zouain, N. A. and Herskovits, J.: An Optimality Criteria Method Based on Slack Variable Concept for Large Scale Structural Optimization. Paper presented at Symposium on Application of Computer Methods in Engineering, University of Southern California, Los Angeles, CA, 1977.

5. Rizzi, P.: Optimization of Multi-Constrained Structures Based on Optimality Criteria. Proceedings AIAA/ASME/SAE 17th Structures, Structural Dynamics, and Materials Conference, AIAA, New York, 1976, pp. 448-462.

6. Fleury, C. and Sander, G.: Structural Optimization by Finite Element. Report NBR-SA-58, Jan. 1978, Aerospace Laboratory, University of Liege, Belgium.

7. Khot, N. S., Berke, L. and Venkayya, V. B.: Comparison of Optimality Criteria Algorithms for Minimum Weight Design of Structures. AIAA Journal, Vol. 16, No. 2, Feb. 1979, pp. 182-190.

8. Kiusalaas, J.: Minimum Weight Design of Structures via Optimality Criteria. NASA TN D-7115, Dec. 1972.

9. Haftka, R. T. and Starnes, J. H., Jr.: WIDOWAC (Wing Design Optimization with Aeroelastic Constraints): Program Manual. NASA TM X-3071, Oct. 1974.

10. Haftka, R. T. and Yates, E. C., Jr.: Repetitive Flutter Calculations in Structural Design. Jour. Aircraft, Vol. 13, No. 7, July 1976, pp. 454-461.

REFERENCES (Concluded)

11. Craig, R. R., Jr. and Erbug, I. O.: Application of a Gradient-Projection Method to Minimum Weight Design of a Delta Wing with Static and Aeroelastic Constraints. Computers and Structures, Vol. 6, June 1976, pp. 529-538.

12. Khot, N. S., Berke. L. and Venkayya, V. B.: Minimum Weight Design of Structures by the Optimality Criterion and Projection Method. AIAA Paper No. 79-0720, presented at AIAA/ASME/ASCE/ AHS 20th Structures, Structural Dynamics, and Materials Conference, St. Louis, MO, April 4-6, 1979.

13. Gwin, L. B. and McIntosh, S. C., Jr.: A Method of Minimum-Weight Synthesis for Flutter Requirements, Part I - Analytical Investigation. AFFDL TR 72-22, Part I, June 1972.

Table 1.- Comparison of exact and finite-difference derivatives for the biconvex sandwich wing with two flutter constraints (c_1,c_2) and a frequency constraint (c_3)

	c_1			c_2			c_3		
DV No.	Exact	Fin.Diff.	Error, %	Exact	Fin.Diff.	Error, %	Exact	Fin.Diff.	Error, %
1	-0.05607	-0.05525	-1.5	-0.04706	-0.04744	0.80	-0.01307	-0.01302	-0.40
2	-0.4731	-0.4676	-1.2	-0.04229	-0.04005	-5.3	-0.3289	-0.3247	-1.3
3	-0.3568	-0.3529	-1.1	-0.01750	-0.01711	-2.2	-0.1408	-0.1399	0.60
4	-0.6148	-0.6241	1.5	-0.1018	-0.09669	-5.0	-0.03803	-0.03546	6.8
5	-0.03523	-0.03413	-3.1	-0.2188	-0.2221	1.5	0.01645	0.01670	1.5
6	0.01648	0.01542	-6.4	-2.758	-2.592	-6.0	-0.006029	-0.005433	-9.9
7	0.005031	0.004908	-2.4	0.002076	0.001642	-21	0.0002239	0.0002575	15.0

Table 2.- Variation of iterations required for convergence with various parameters in SLACK for test case of figure 5.

ε_v	δ_c	K	$(\nu)_{max}$
0.001	0.5	2.0	32
0.01	0.5	2.0	17
0.01	0.1	2.0	75
0.01	0.1	1.5	64
0.01	0.1	1.0	52

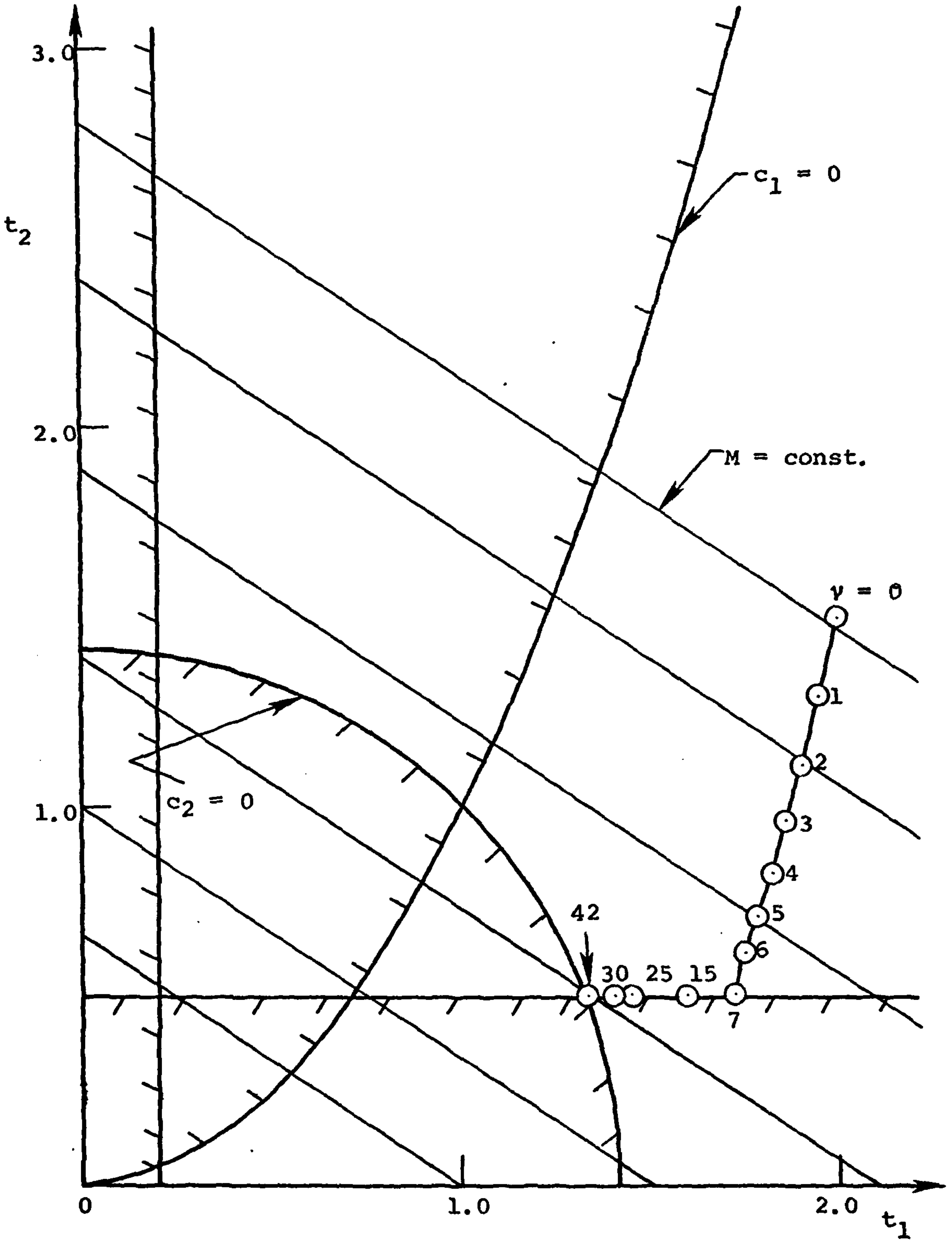

Figure 1.- SLACK iteration history in design space for 2-DV test case, With $\delta_t = 0.1$, $\delta_c = 1.0$, $t_\ell = 0.9$, $d = 2.5$, $K = 2.0$, and $\epsilon_v = 0.01$.

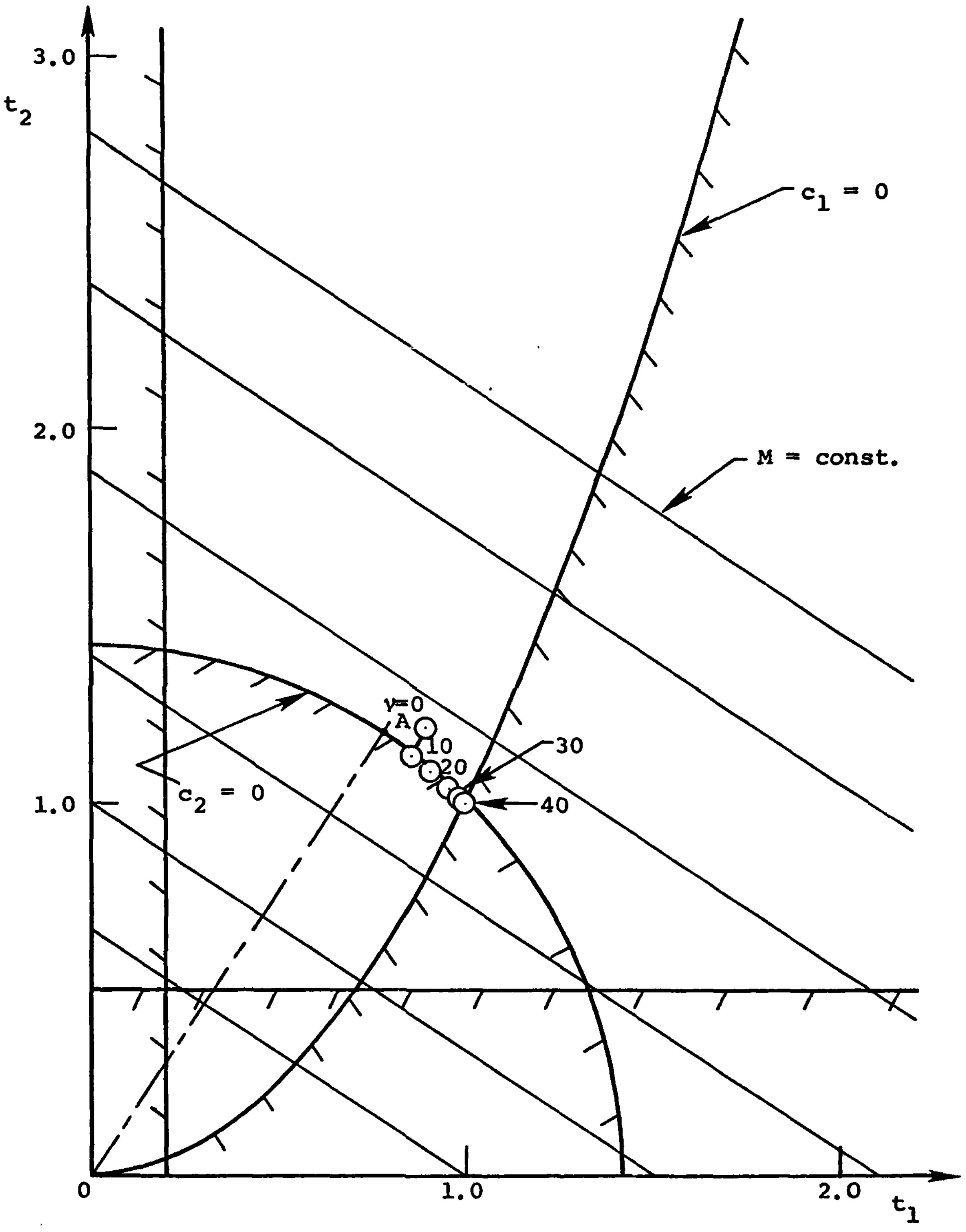

Figure 2.- SLACK iteration history in design space for 2-DV test case, with $\delta_t = 0.5$, $\delta_c = 1.0$, $t_\ell = 0.9$, $d = 2.5$, $K = 2.0$, and $\epsilon_v = 0.001$.

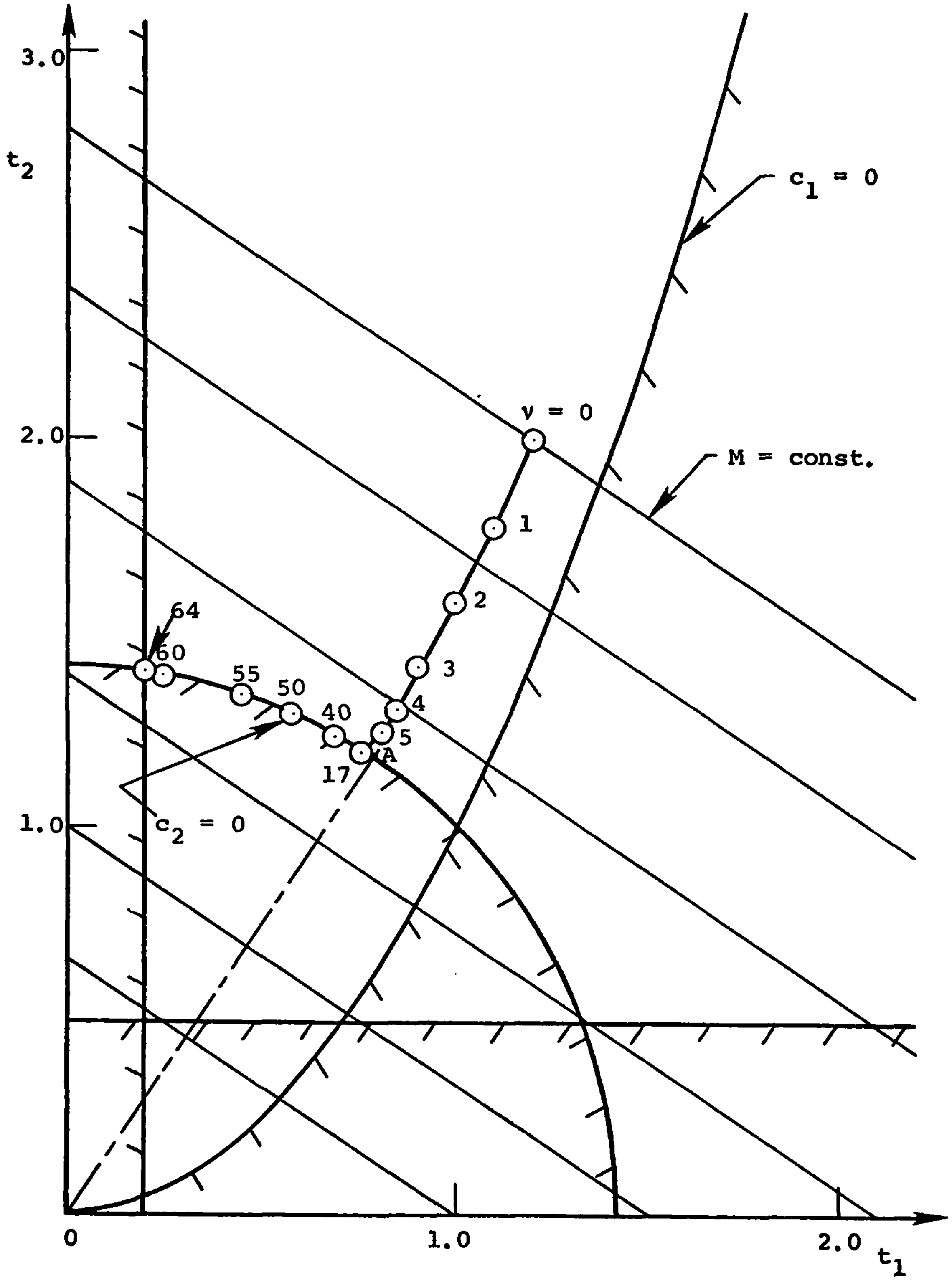

Figure 3.- SLACK iteration history in design space for 2-DV test case, with $\delta_t = 0.1$, $\delta_c = 0.5$, $t_l = 0.9$, $d = 2.5$, $K = 2.0$, and $\epsilon_v = 0.001$.

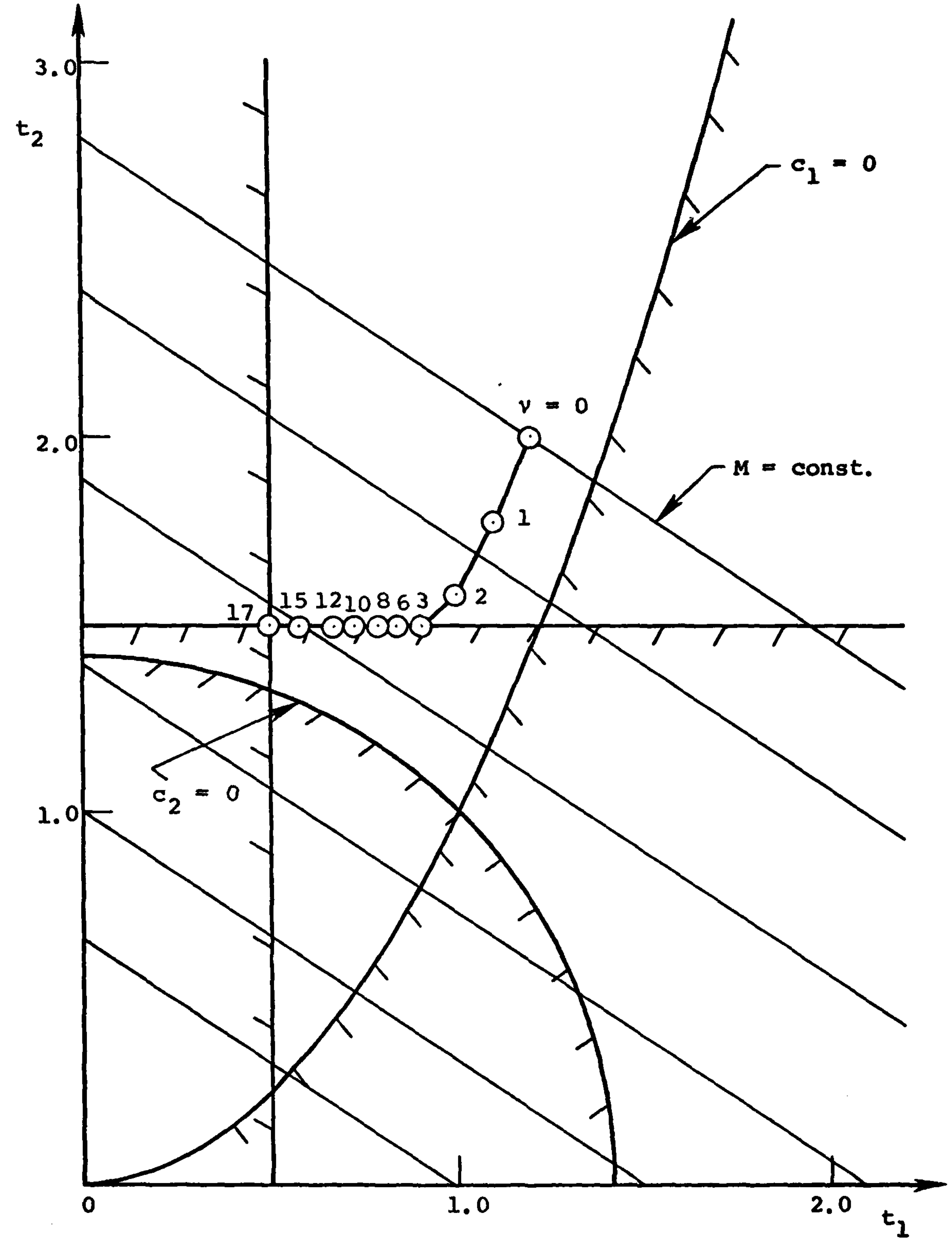

Figure 4.- SLACK iteration history in design space for 2-DV test case, with $\delta_t = 0.5$, $\delta_c = 0.5$, $t_\ell = 0.9$, $\bar{d} = 2.5$, $\bar{K} = 2.0$, and $\epsilon_v = 0.001$.

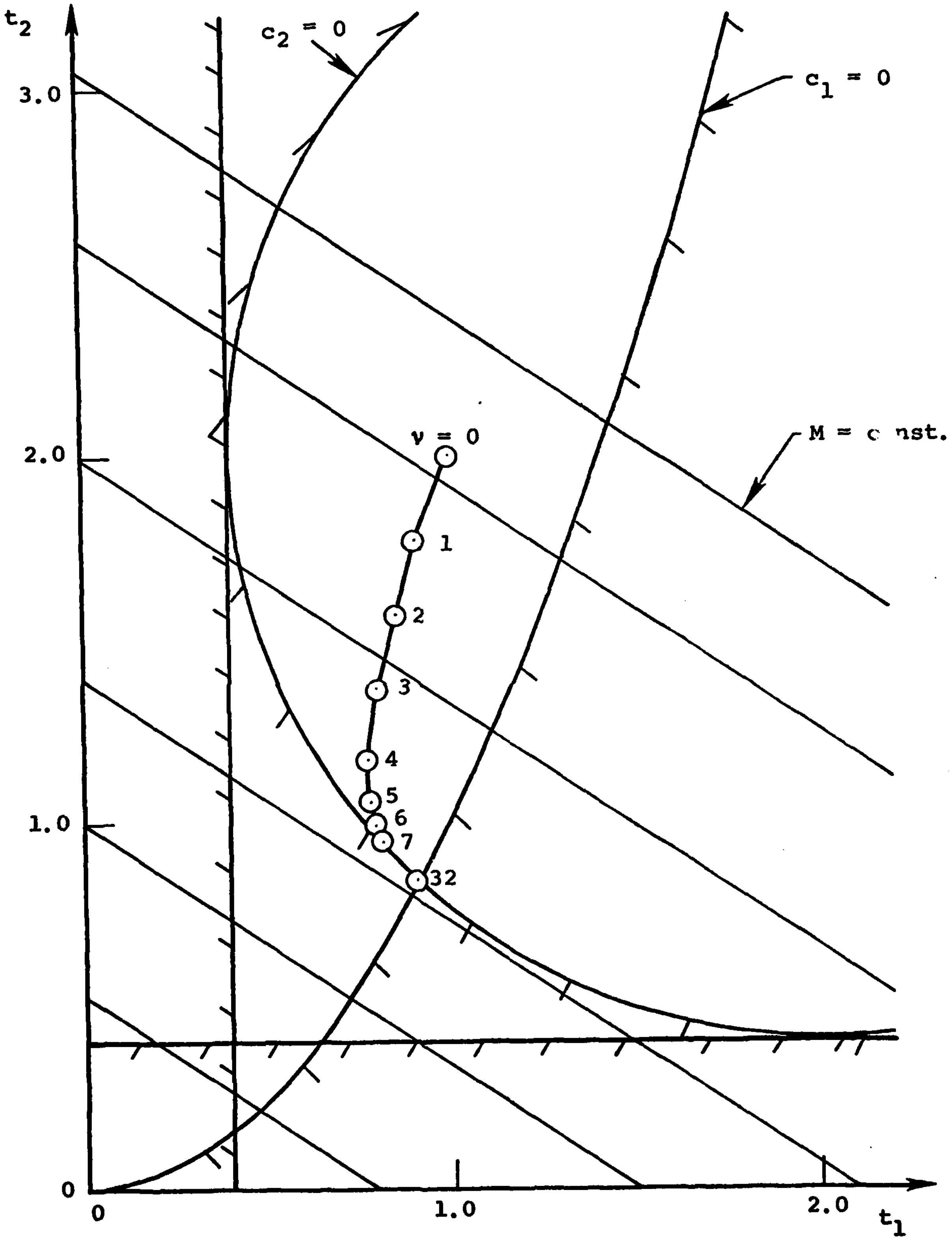

Figure 5.- SLACK iteration history in design space for 2-DV test case, with $\delta_t = 0.1$, $\delta_c = 0.5$, $t_\ell = 0.9$, $d = 2.5$, $\bar{K} = 2.0$, and $\epsilon_v = 0.001$.

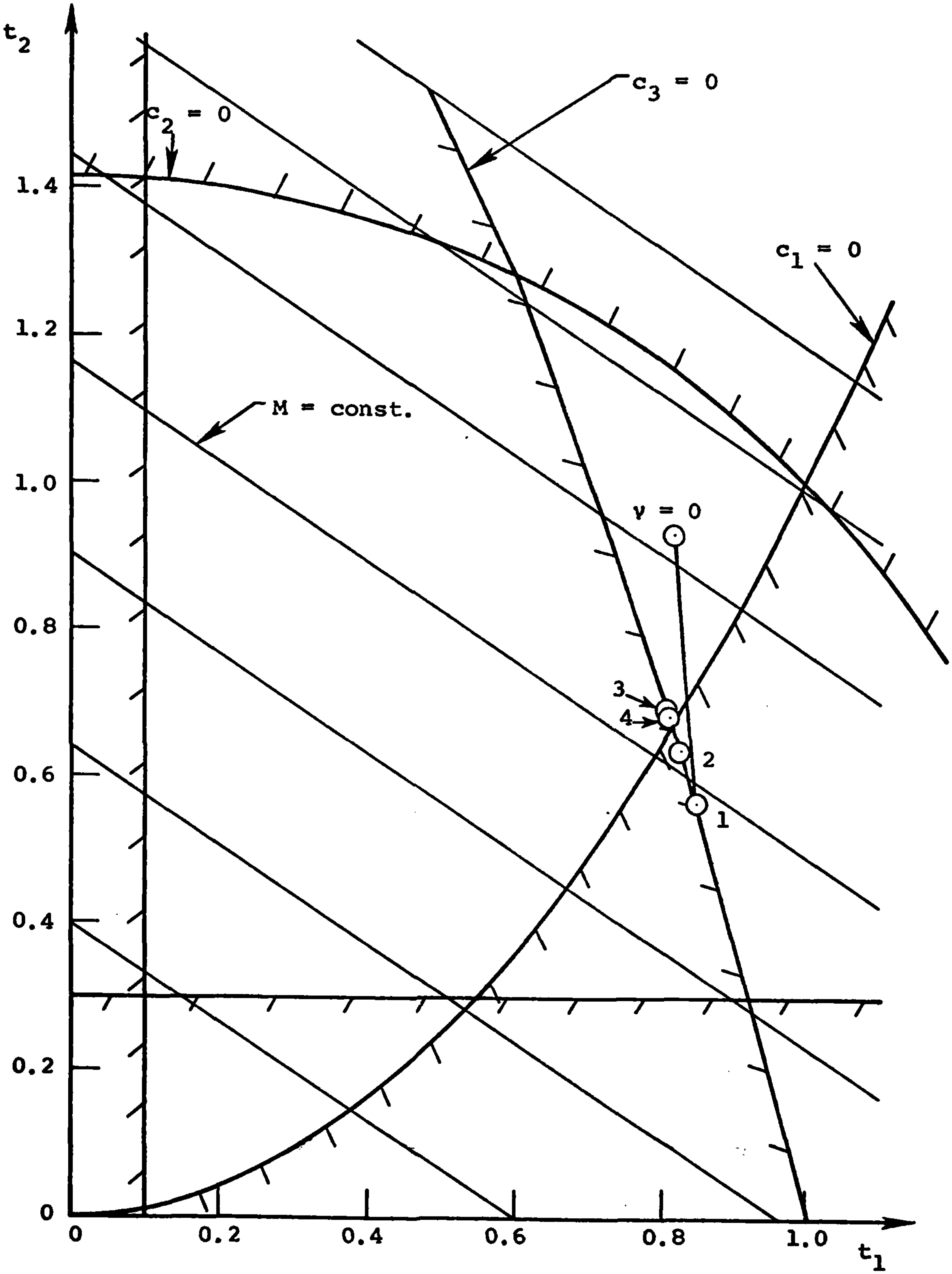

Figure 6.- GAUSS iteration history in design space for 2-DV test case, with $\alpha^{(0)} = 0.5$, $\alpha_x = 1.05$, $\gamma^{(0)} = 0.35$, $\delta_x = 1.05$, $\epsilon_v = 0.05$, and $\epsilon_c = 0.05$.

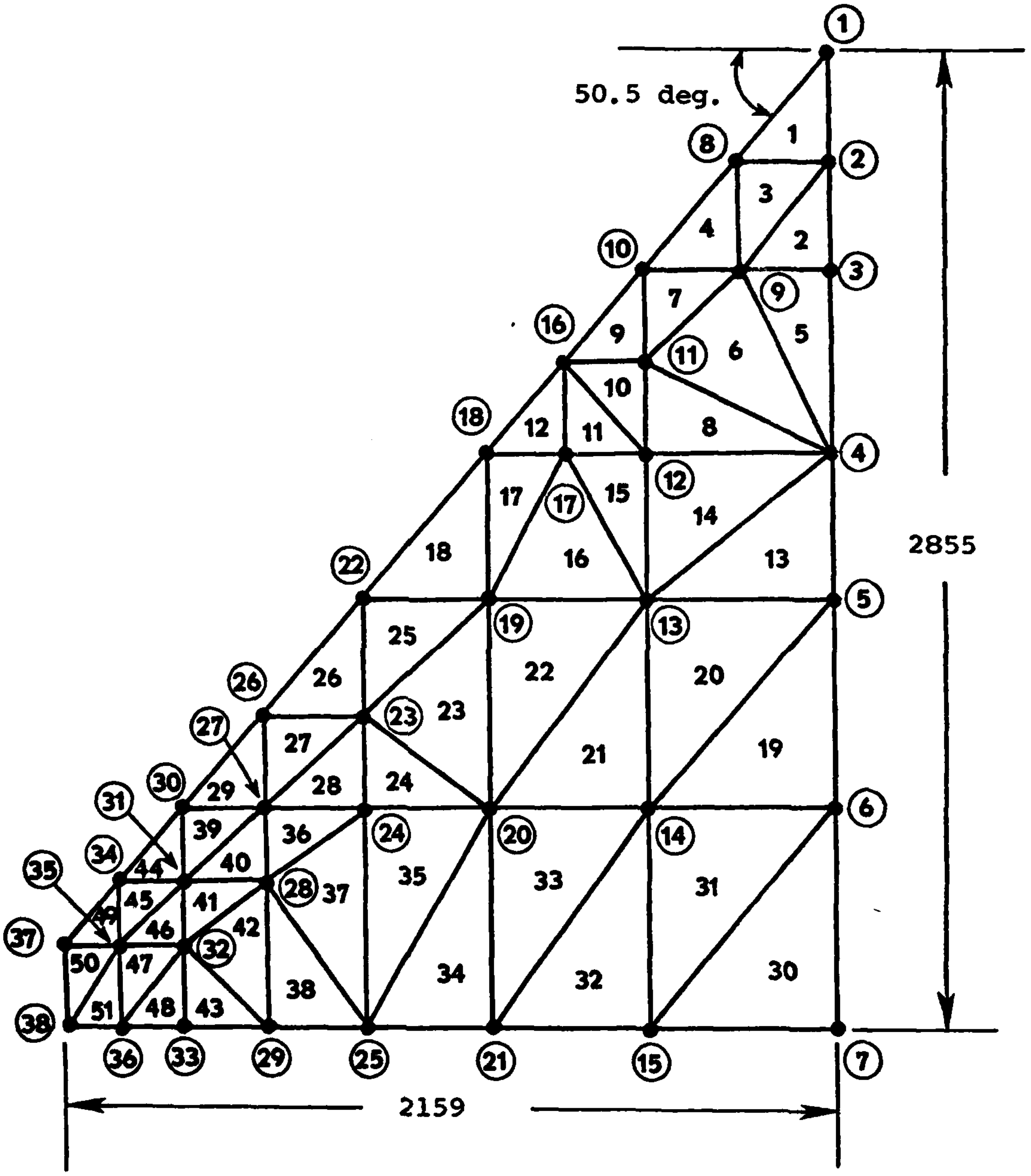

Figure 7.- Upper-surface node-point and cover-panel numbering for biconvex sandwich wing. All dimensions in cm.

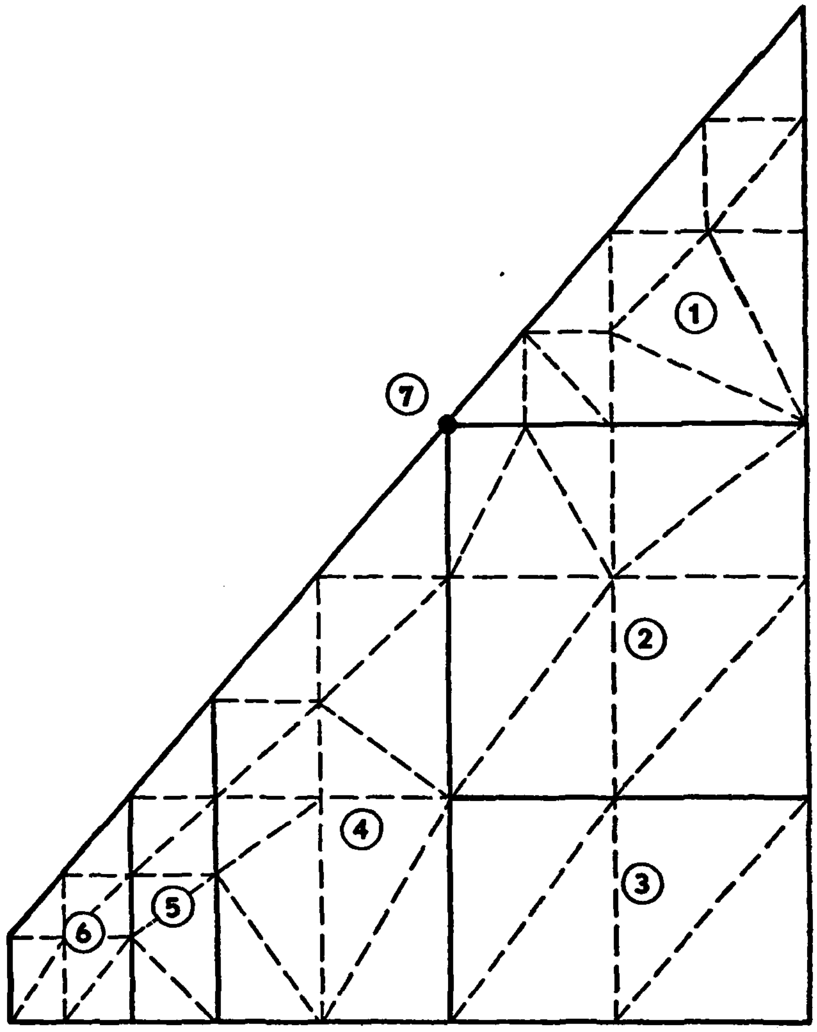

Figure 8.- Design-variable linking for biconvex sandwich wing.

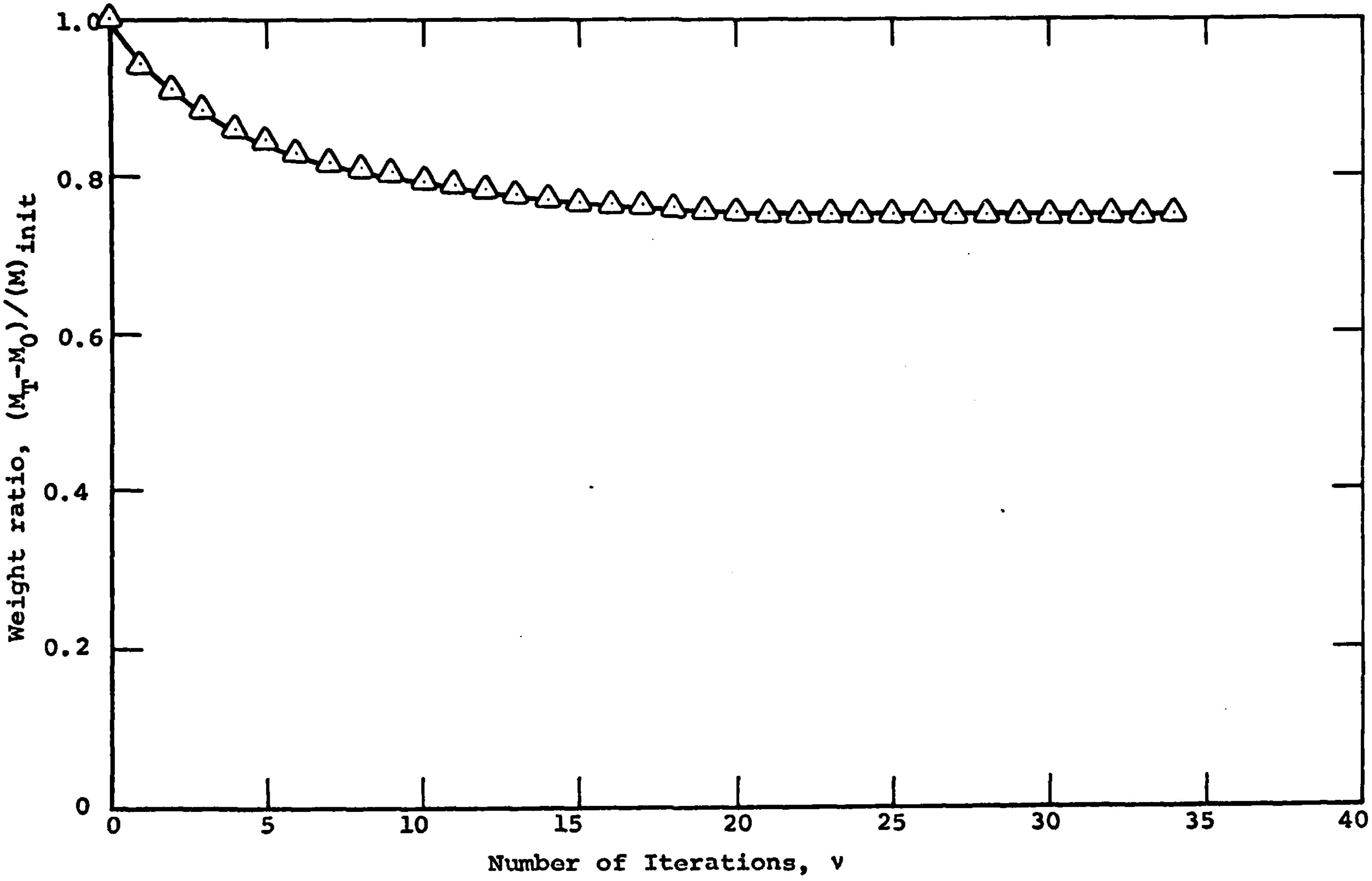

Figure 9.- SLACK weight iteration history for biconvex wing with two frequency constraints.

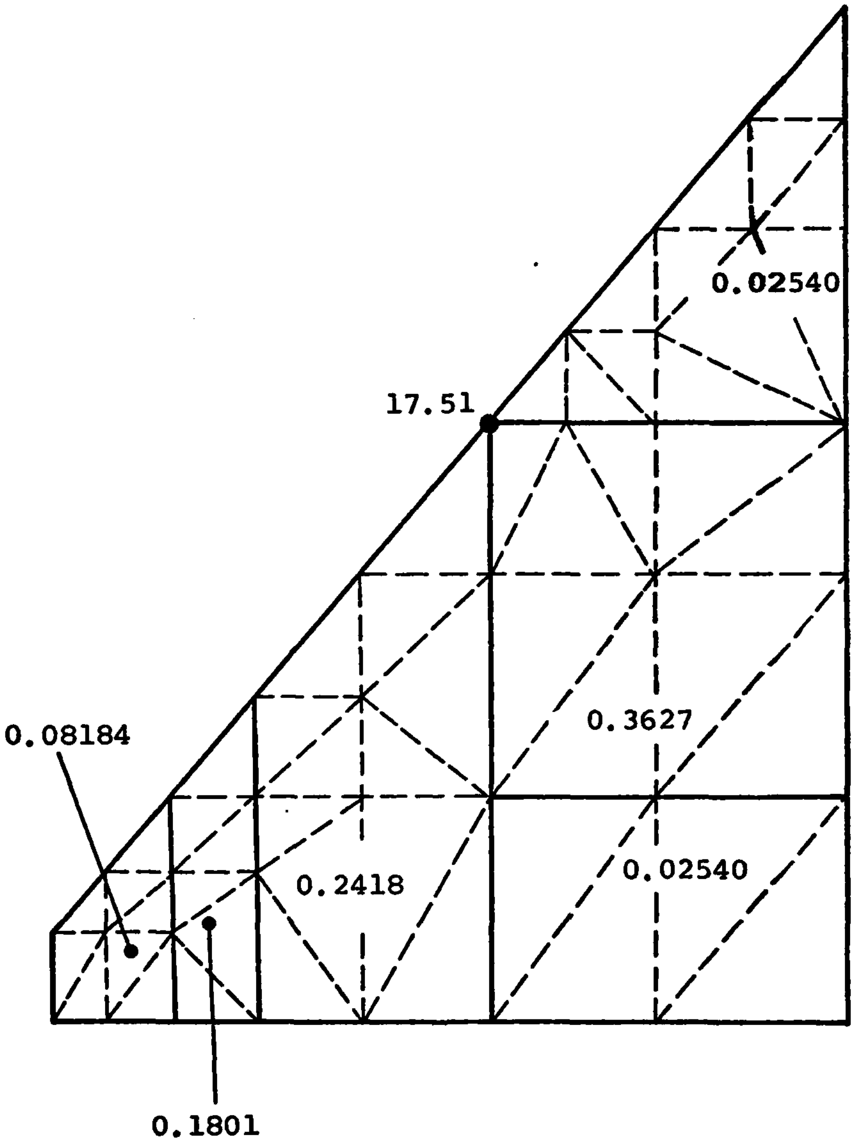

Figure 10.- Optimal values of design variables (one surface) for biconvex wing with two frequency constraints. Skin thicknesses in cm; tuning-mass value in kg.

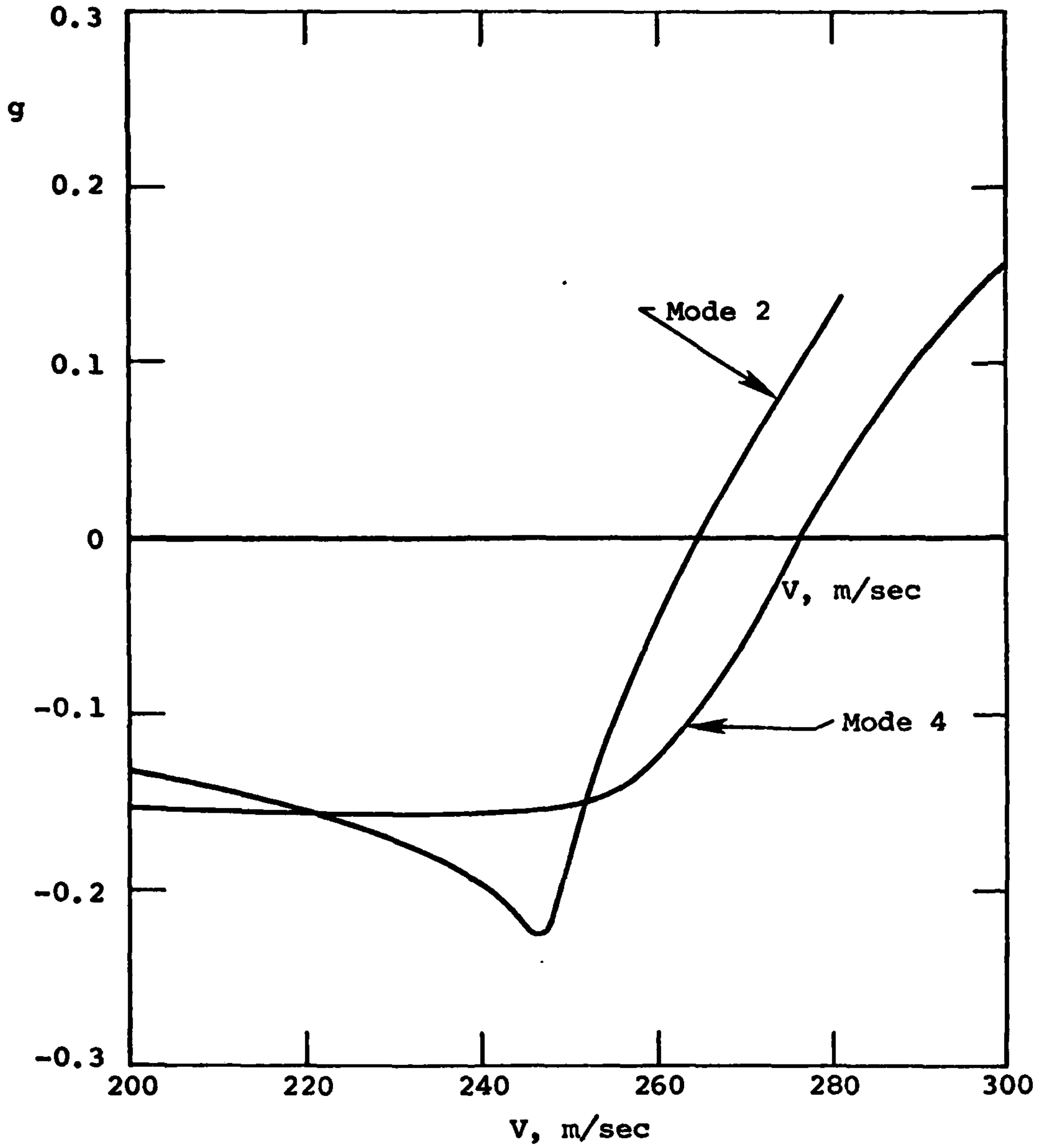

Figure 11.- Critical V-g curves for biconvex wing with flutter constraint after 5th iteration in SLACK.

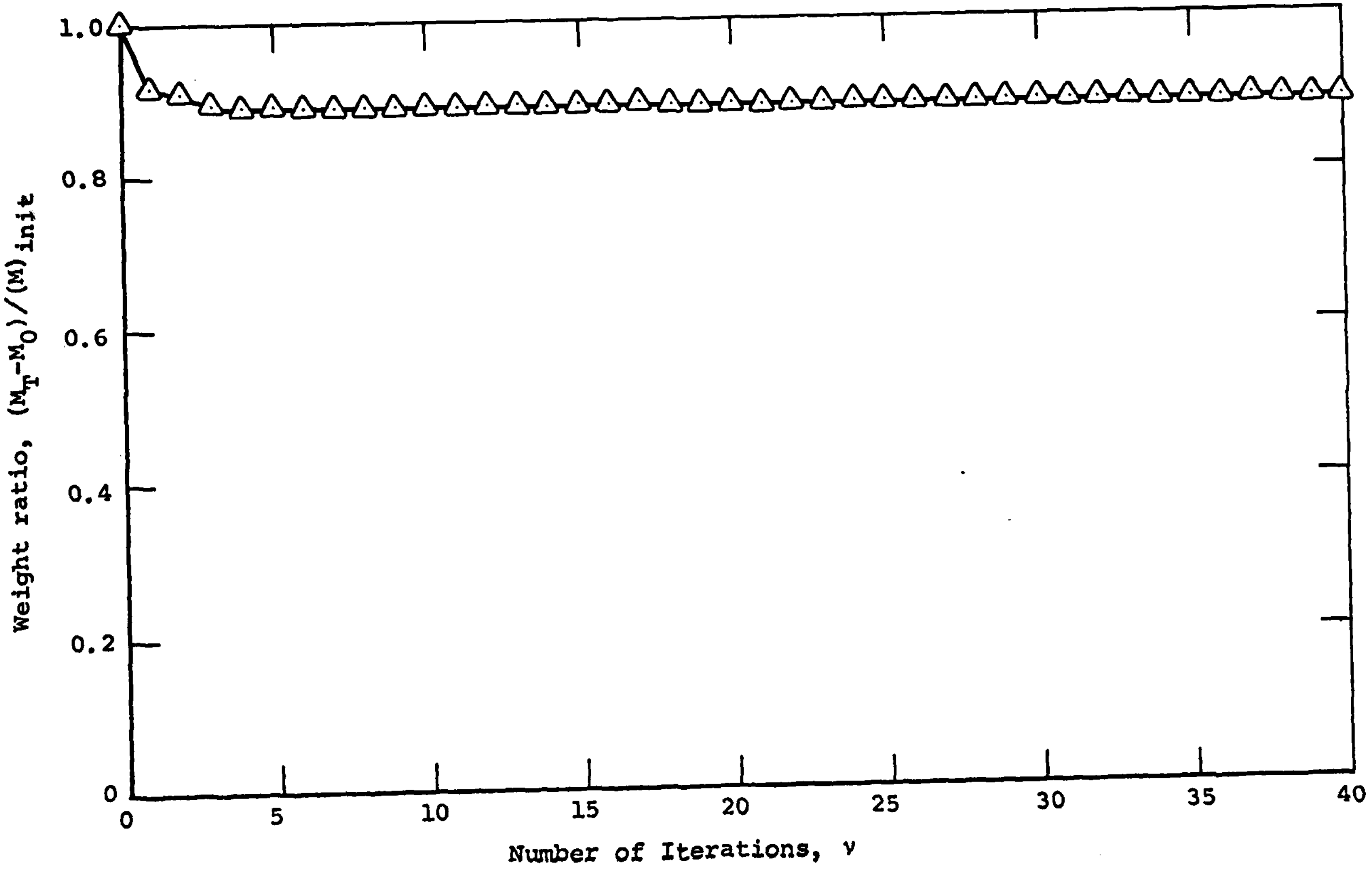

Figure 12.- SLACK weight iteration history for biconvex wing with two flutter constraints.

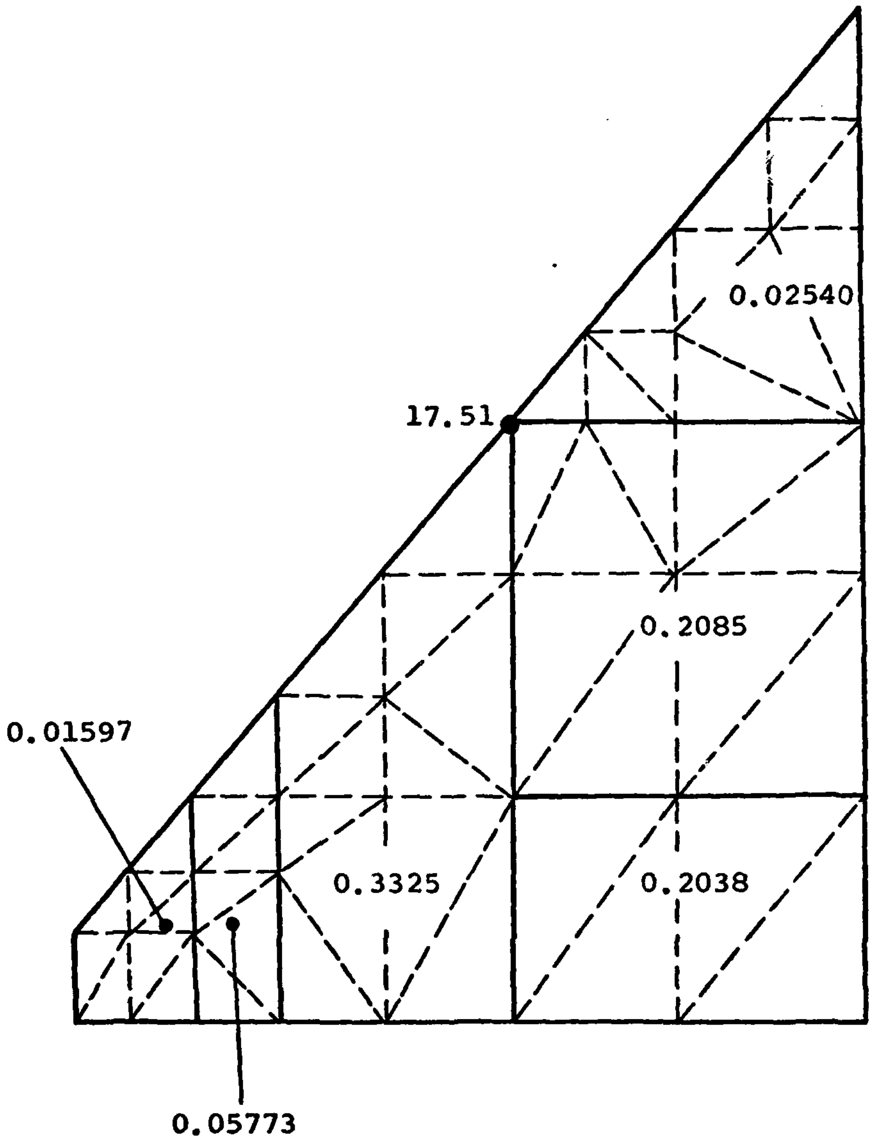

Figure 13.- Optimal values of design variables (one surface) for biconvex wing with two flutter constraints. Skin thicknesses in cm; tuning-mass value in kg.

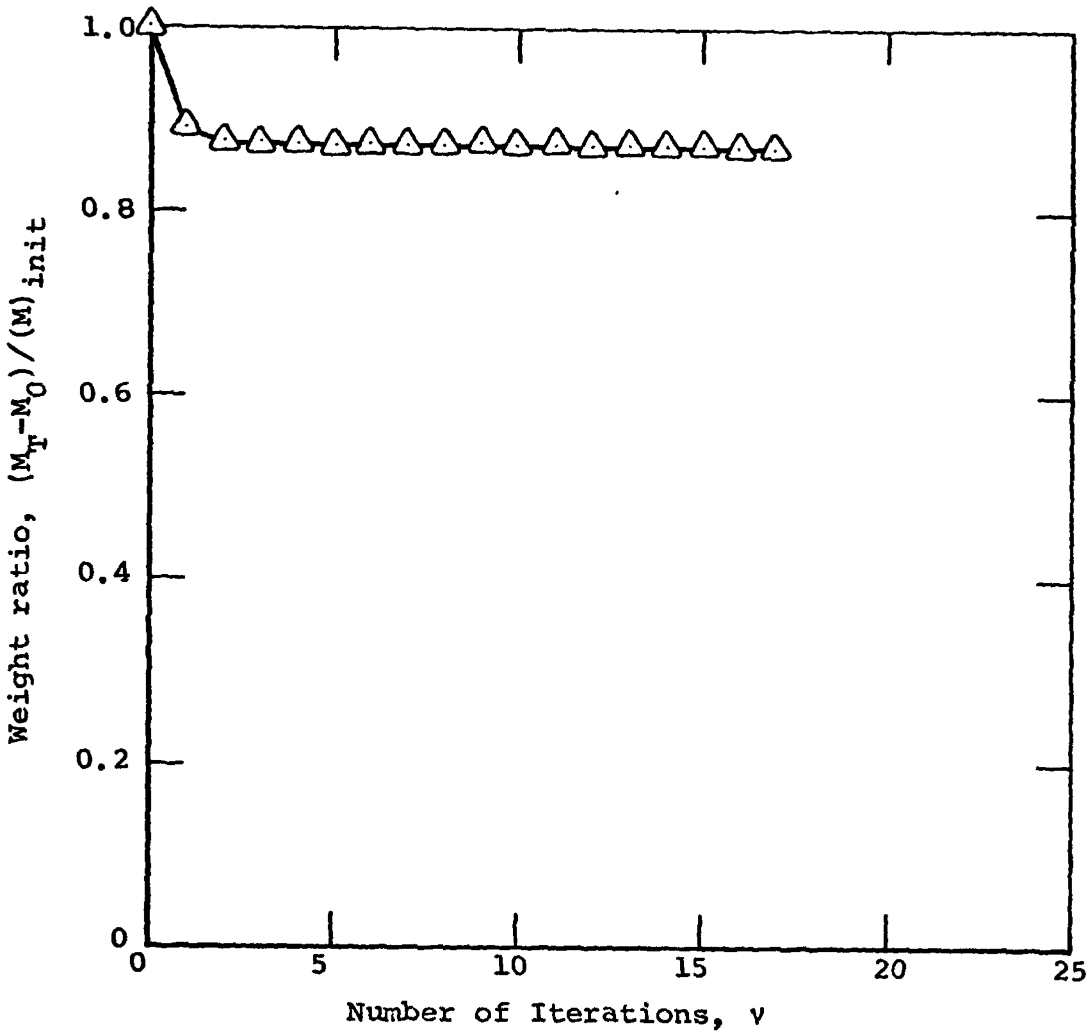

Figure 14.- SLACK weight iteration history for biconvex wing with two flutter constraints and one frequency constraint.

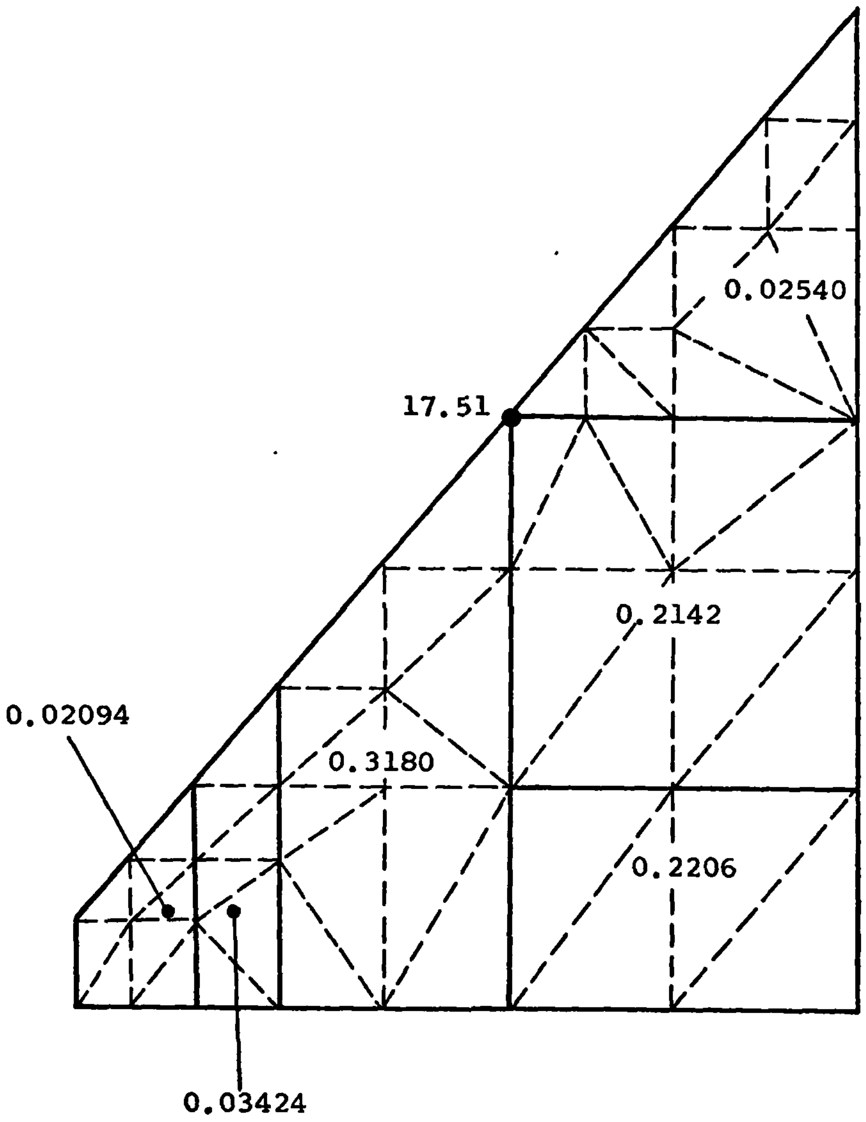

Figure 15.- Optimal values of design variables (one surface) for biconvex wing with two flutter constraints and one frequency constraint. Skin thicknesses in cm; tuning-mass value in kg.

NOMENCLATURE

a_i	Wright coefficient - see equation (1)
A_c	Preliminary set of active or violated constraints
$\overline{A}_c$	Set of active or violated constraints as identified by Gauss-Seidel iteration
A_v	Set of active design variables
b	Reference length
c, c_j	Inequality constraint function; jth inequality constraint function
c'_j	jth equality constraint function, $c_j + z_j^2$
C_i	Redesign factor on design variables
d	Limiting value on $\|\alpha-1\|$ - see equation (33)
D_j	Redesign factor on slack variables
E	Young's modulus
EA	Product of Young's modulus and cross-sectional area
[GA]	Generalized aerodynamics matrix
[GM],[GK]	Generalized mass and stiffness matrices, respectively
$[GM_i],[GK_i]$	Derivatives of generalized mass and stiffness matrices with respect to design variable t_i
k	Reduced frequency, $\omega b/V_\infty$
K	Step-size parameter for z_j
$[K_i]$	Derivative of discrete stiffness matrix with respect to design variable t_i
M_o	Mass not available for optimization
M_T	Total mass
M	Number of design variables
M_∞	Mach number

NOMENCLATURE (Continued)

N	Number of constraints
N_A	Number of active design variables
P_ℓ	Set of design variables predicted to reach a lower bound during an iteration
$\bar{P}_\ell$	Set of all lower-bound passive variables
P_u	Set of design variables predicted to reach an upper bound during an iteration
$\bar{P}_u$	Set of all upper-bound passive variables
t_i	ith design variable
u_r	Displacment being constrained; rth displacement in column of node-point displacements
t_u, t_ℓ	Parameters for determining P_u and P_ℓ respectively
$\bar{v}_i$	Convergence function - see equation (4)
v_i	Normalized convergence function, $\bar{v}_i/a_i$
V_∞	Free-stream speed
z_j	jth slack variable
α	Step-size parameter
α_x	Multiplier to determine α at each iteration - see equation (48)
β_j, β_{kj}	Functions used in determining Lagrange multipliers - see equations (21) and (22)
$\bar{\beta}_{kj}$	Augmented function - see equation (25)
δ_t	Step-size parameter used in determining $\lvert\alpha-1\rvert$ - see equation (28)
δ_c	Constraint-change parameter used in determining $\lvert\alpha-1\rvert$ - see equation (30)
$\bar{\delta}_c$	Constraint tolerance parameter used to define A_c - see equation (41)

NONMENCLATURE (Concluded)

δ_x	Multiplier to determine $\overline{\delta}_c$ at each iteration - see equation (49)
ε_v	Convergence parameter for v_i
ε_c	Convergence parameter for c_j
λ_j	jth Lagrange multiplier
ν	Iteration number
$\overline{\nu}$	Poisson's ratio
ρ	Mass density
ω, ω_r	Frequency; rth natural frequency
$\overline{\Omega}$	Eigenvalue from aeroelastic equations
$(\ )^{\nu}$	Quantity at νth iteration
$(\ )_{des}$	Desired quantity